# ESCUELA POLITECNICA NACIONAL

## FACULTAD DE INGENIERIA ELECTRICA

"PROGRAMA SEMIGRAFICO PARA ANALIZAR ESTABILIDAD
DE SISTEMAS NO LINEALES
USANDO LA FUNCION DESCRIPTIVA"

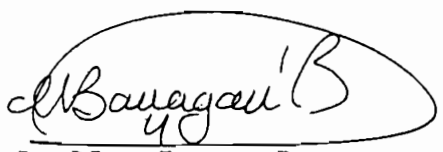
POR

WILMER PATRICIO CARDENAS BURBANO

TESIS PREVIA A LA OBTENCION DEL TITULO DE INGENIERO
EN ELECTRONICA Y CONTROL

QUITO, OCTUBRE DE 1994

CERTIFICO QUE EL PRESENTE TRABAJO HA SIDO REALIZADO EN SU TOTALIDAD POR EL SR. WILMER PATRICIO CARDENAS BURBANO



Ing. Marco Barragán B.

DIRECTOR DE TESIS

## AGRADECIMIENTO

Un sincero agradecimiento al Ing. Marco Barragán Bedoya por su acertada dirección en esta tesis, ya que sin su apoyo e interés no hubiera sido posible este trabajo.

## DEDICATORIA

### A MIS PADRES

Que con su comprensión y amor, me impulsaron a entregar toda mi capacidad, que, supieron sembrar en mi el anhelo de superación.

Página

CAPITULO II ANALISIS DE PROGRAMAS REALIZADOS


2.1 Análisis para graficar -1/N G(jw)
y la superposición ..... 71

2.1.1 Ingreso de la Función de Transferencia .. 71

2.1.2 Ingreso de la Función Descriptiva ..... 78

2.1.3 Graficación ..... 90

2.2 Presentación de resultados ..... 95

2.3 Estabilidad ..... 101

2.4 Diseño ..... 105

2.5 Simulación ..... 109


CAPITULO III EJERCICIOS


3.1 Ejercicios para comprobar resultados ..... 118

3.2 Ejercicios para probar limitaciones
del programa ..... 136

3.3 Ejercicios para probar la calidad del programa ... 141

Página

CAPITULO IV CONCLUSIONES

4.1 Conclusiones ..... 144

BIBLIOGRAFIA ..... 151

ANEXOS

ANEXO A MANUAL DE UTILIZACION DEL PROGRAMA ..... 156

ANEXO B LISTADO DE PROGRAMAS ..... 170

# INTRODUCCION

# INTRODUCCION

"Un *servomecanismo* puede ser definido como un elemento que mide la diferencia entre un estado real y un estado deseado, y usa la diferencia para conducir el estado actual hacia el deseado,"

Esta definición se la representa en la figura siguiente:

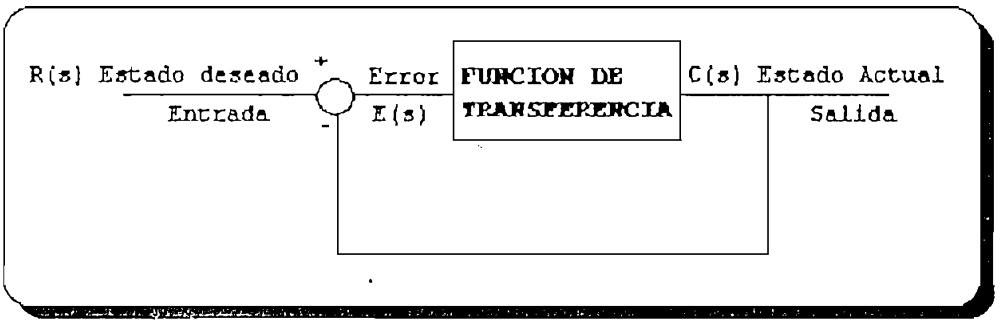



*Fig. i.1 Representación de un servomecanismo*

En la figura i.2 se muestra en forma más general un diagrama en lazo cerrado indicando las funciones de transferencia tanto de la planta como de la realimentación.

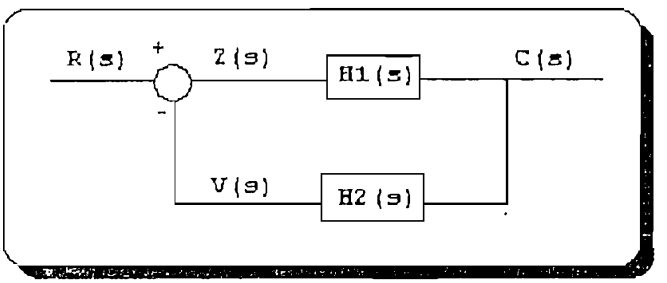



*Fig i.2 Sistema en lazo cerrado*

Como se observa en este caso Z no es la diferencia entre la entrada y la salida, por lo tanto no corresponde al error normal, pero si al llamado *Error actuante*. En este caso las funciones de transferencia $H_1$ y $H_2$ son de la planta y del lazo de realimentación respectivamente, y corresponden a:

$$H_1(s)=\frac{C(s)}{Z(s)} \quad y \quad H_2(s)=\frac{V(s)}{C(s)}$$

De la figura i.2 se desprende que

$$Z(s) = R(s) - V(s)$$

Pero si se reemplaza la definición tanto de $H_1(s)$ como de $H_2(s)$, se llega a:

$$\frac{C(s)}{H_1(s)}=R(s)-H_2(s)C(s)$$

Resolviendo esta ecuación se llega a la función de transferencia equivalente de lazo cerrado.

$$\frac{C(s)}{R(s)}=\frac{1}{1/H_1(s)+H_2(s)}=\frac{H_1(s)}{1+H_1(s)H_2(s)}$$

Definiendo a $H_1(s)$ = A , ganancia en lazo abierto, entonces la expresión de la función de transferencia en lazo cerrado queda de la siguiente manera

$$\frac{C(s)}{R(s)}=\frac{A(s)}{1+A(s)} \qquad H_2(s) = 1;$$

Esta técnica es utilizada para aquellos sistemas que mantienen un comportamiento lineal, en donde también se utiliza muchas veces el análisis en el dominio del tiempo. Cuando se trata de sistemas no lineales, es más factible emplear técnicas en el dominio de la frecuencia real w.

Lo anterior es importante pues dentro de una industria o un proceso cualquiera pueden existir elementos no lineales, o procesos no lineales, que muchas veces se los aproxima por medio de ecuaciones lineales, para con ello obtener una simplificación matemática, que dependiendo del caso puede alejarse de la realidad. La forma de comprobar que dichas aproximaciones sean aceptables es comparando con los resultados que se pueden obtener experimentalmente.

El tratar de mantener un comportamiento lineal de un sistema durante todo su intervalo de funcionamiento, puede ser muy difícil de conseguir, y el resultado no sería muy práctico, debido a los elevados costos que involucraría el conseguir elementos completamente lineales, pudiéndose conseguir mejores resultados con un sistema no lineal adecuadamente diseñado.

Es útil entonces indicar que un sistema no lineal es aquel que es representado por ecuaciones no lineales como por ejemplo:

$$y = \text{sen } x$$

$$y = x^2$$

También se puede tener ecuaciones diferenciales no lineales de la siguiente manera:

$$\frac{d^2x}{dt^2} + \left(\frac{dx}{dt}\right)^2 + x = 0$$

En la práctica muchos sistemas llamados lineales, se comportan como tales en un cierto rango de operación; esto aparece por ejemplo en elementos que tienen un comportamiento lineal mientras no lleguen a saturarse, ya que cuando la entrada sea muy grande en un momento dado, por efecto de la saturación, la salida se verá afectada.

También pueden presentarse casos en los cuales por las características propias del sistema, este muestre cierto ciclo de histéresis.

Una de las características más importantes que presentan los sistemas no lineales, es la dependencia en el comportamiento de la respuesta del sistema, tanto de la magnitud como del tipo de entrada, es decir si a un sistema se le cambia la amplitud de entrada de un escalón, la respuesta puede ser diferente, y no proporcional a la primera amplitud.

Otra característica importante que hay que destacar en los sistemas no lineales, es que en este tipo de sistemas obviamente no es aplicable el principio de superposición, y por ello el llegar a una solución de un sistema no lineal resulta muy complicado, de ahí que muchas veces se buscan sistemas lineales equivalentes más precisos, para representar a los no

lineales, aunque dichos sistemas solo dan buenos resultados en un cierto rango de operación como ya se anotó.

En un sistema se pueden tener dos tipos de alinealidades, en general, primero las que existen en forma inherente en el sistema, y segundo, aquellas alinealidades puestas intencionalmente en el proceso.

*Alinealidades inherentes*

Entre las principales alinealidades inherentes se tienen:

1. Saturación
2. Zona muerta
3. Histéresis
4. Juego
5. Fricción estática, fricción de Coulomb y otras fricciones no lineales
6. Resorte no lineal
7. Compresibilidad del fluido.

*Alinealidades intencionales*

Este tipo de alinealidades se las introduce en un sistema con la finalidad de mejorar el comportamiento del mismo, para tomar alguna acción de control, como también para simplificar o abaratar su construcción. Un

ejemplo muy simple de este tipo de alinealidades es la utilización de un relé para conectar o desconectar una cierta parte de un proceso.

Debido a que las ecuaciones diferenciales no lineales no son simples de resolver, ya que no se posee un método general de resolución y más aún se podrían obtener soluciones exactas sólo en aquellas ecuaciones no lineales simples; entonces se prefiere aplicar técnicas como por ejemplo el uso de la *Función Descriptiva*, el mismo que es un sistema de linealización equivalente, válido no solamente para señales pequeñas.

Muchas veces lo que interesa es efectuar un análisis de estabilidad antes que resolver exactamente el sistema no lineal, por ello es más importante establecer criterios de estabilidad, que resultaría más simple que resolver las ecuaciones diferenciales no lineales.

Entre algunos métodos que dan información de la estabilidad de un sistema no lineal, se presentan:

- *Método del plano de fase*, el mismo que da información tanto respecto a estabilidad como a respuesta en el tiempo, pero presenta el inconveniente de que sólo es útil para sistemas de primero y segundo orden.

- *El segundo método de Liapunov*, que puede ser aplicado a cualquier sistema no lineal, pero su dificultad está en hallar las funciones de Liapunov cuando se trata de sistema complicados.

- *Método de la Función Descriptiva*, que se empleará en el presente trabajo, en donde se hace una representación gráfica de los resultados para realizar el análisis de los sistemas de control no lineales.

Contenido de la Tesis.-

El presente trabajo de tesis esta constituido básicamente por tres capítulos.

- El primer capítulo es todo un análisis de las bases teóricas que sirvieron para el desarrollo de este programa, aquí se abordan criterios de estabilidad como el de Nyquist, que es el que se lo aplica principalmente en este trabajo

- El capítulo segundo explica los diversos algoritmos empleados en el programa, explicando para ello sus diagramas de flujo, de todo lo que tiene que ver con graficación y con respecto a la obtención y presentación de resultados.

- En el capítulo tres se realizan una serie de ejercicios que tienen como objeto probar la validez del programa, de los cuales varios de ellos han sido tomados de libros, y otros de ejercicios realizados en clase, indicando en cada uno de ellos los resultados que se obtienen, y los posibles problemas que pueden aparecer en el programa.

- Por último se presenta una serie de conclusiones obtenidas en el desarrollo de este programa, así como también de los problemas afrontados y de los sistemas analizados.

- Como complemento a lo anterior se incluyen dos anexos, el primero es un pequeño manual de usuario, el mismo que tiene como objetivo el indicar la manera de sacarle el mejor provecho a este programa. Mientras que el ANEXO B presenta un listado total del programa, el

mismo que ha sido dividido en 12 grupos, aparentemente independientes unos de otros, estos grupos de programa son:

INICIO.CPP.- En donde se enlazan todos los demás grupos, manejando además aquí todo lo que tiene que ver con la presentación de pantallas y menús desplegables del programa

PRES_OUT.CPP.- El cual tiene los programas necesarios para realizar la presentación de resultados, es decir para grabar o imprimir los resultados en un archivo.

FUN_TRAN.CPP.- Aquí se tienen los programas que sirven para el ingreso y la determinación de la respuesta de frecuencia de la planta tanto compensada como sin compensación.

FIGURAS.CPP.- Constituye una biblioteca gráfica, en donde se encuentran almacenadas las gráficas de cada tipo de alinealidad que se puede usar.

FUNDESC.CPP.- Son los programas para el ingreso como también para la determinación de la función descriptiva de cada una de las posibles alinealidades a usarse.

VER_FUN.CPP.- Almacena la programación que permite presentar en pantalla cual es tanto la función de transferencia de la planta, la compensación, y la alinealidad que se encuentra usándose en cualquier momento de ejecución del programa.

GRAFICAR.CPP.- Son listados de programas de la graficación de las curvas G(jw) y -1/N, con y sin compensación.

PTOS_CRUC.CPP.- Lo constituyen los programas que se usan para encontrar las coordenadas de los puntos de cruce de las curvas -1/N y G(jw)

GR_TIEMPO.CPP.- Guarda los programas usados para graficar la respuesta en el tiempo del sistema.

FILEVIEW.CPP.- Es el programa que permite abrir archivos tipo ASCII y presentarlos en pantalla.

RELOJ.CPP.- Programa que lee la hora del reloj interno de la computadora y lo presenta en la pantalla en la esquina superior derecha.

CALC.CPP.- Esta formado por varios programas que manejan una calculadora con que dispone el programa.

# CAPITULO I

## FUNDAMENTOS TEORICOS

# CAPITULO I

## 1.1 FUNCION DESCRIPTIVA

Este método permite realizar un estudio de estabilidad de sistemas de control simples, no lineales, en el dominio de la frecuencia.

El método de la función descriptiva es útil para determinar la estabilidad de un sistema no lineal, pero es necesario tener un poco de cuidado cuando este método se lo aplica en la optimización y diseño de sistemas.

Un sistema con un elemento no lineal es como el representado en la figura 1.1

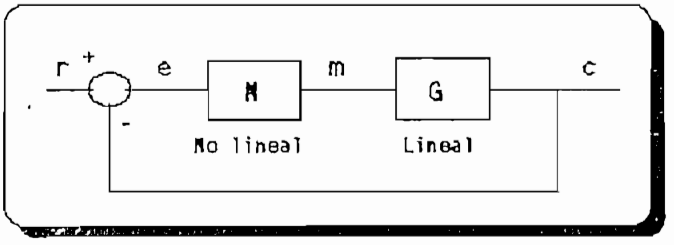



*Fig. 1.1 Sistema lineal con un elemento no lineal*

En la figura 1.1, G es la función de transferencia de la parte lineal reducida, que no depende de la amplitud de la señal de entrada, mientras que N es la función descriptiva de la parte no lineal reducida la cual

depende siempre de la amplitud y en ciertos casos de la frecuencia de la entrada.

Cuando a la entrada de un elemento no lineal ingresa una onda sinusoidal de determinada frecuencia, la salida no es una señal sinusoidal, sino una señal que posee una fundamental más un cierto número de armónicas, pero al hacer un análisis por medio de la función descriptiva sólo se considera importante la componente fundamental; esto es frecuentemente válido en sistemas de control, debido a que primeramente la amplitud de las componentes armónicas de la señal obtenida a la salida del elemento no lineal son de menor amplitud que la fundamental, y además por que la mayor parte de estos sistemas de control son filtros pasabajos, los mismos que van a atenuar mucho más las componentes armónicas dejando pasar únicamente la componente fundamental, por esta razón dicha simplificación se la puede aplicar, ya que se define a la *Función Descriptiva o Función sinusoidal Descriptiva*, de un elemento no lineal como la relación compleja entre la componente armónica fundamental de la salida respecto a la entrada, de la siguiente manera:

$$N=\frac{Y1}{X}\phi 1 \qquad (1.1)$$

Donde:

N = Función Descriptiva

X = Amplitud de la sinusoide de entrada

Y1 = Amplitud de la componente armónica fundamental de la salida

$\phi 1$ = Desplazamiento de fase de la componente armónica fundamental de la salida

A la razón de la amplitud de la función descriptiva se la llama *Ganancia del sistema* y al ángulo de fase $\phi$ se lo denomina *cambio de fase* o desplazamiento de fase del sistema.

Para comparar lo que ocurre con sistemas lineales y no lineales, se muestra en la figura 1.2 un diagrama de bloques de un sistema lineal en estado estable.

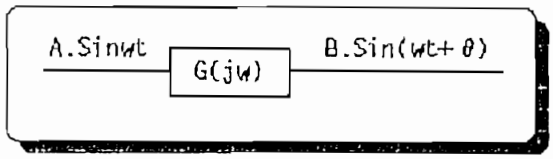



*Fig. 1.2 Diagrama de bloques de un sistema lineal*

Para este caso se define la llamada función de transferencia, de la siguiente manera:

$$G(jW)=\frac{B}{A}e^{j\theta} \qquad (1.2)$$

La ecuación anterior resulta similar a la definición de la función descriptiva (ecuación 1.1)

La ecuación 1.2 sería equivalente para un sistema no lineal en el que la salida sea sinusoidal, pero como ya se indicó la salida de un sistema no lineal no es sinusoidal, pero sí posee una fundamental sinusoidal con un cierto número de armónicos, por lo tanto una expresión más correcta sería considerar la salida de la siguiente manera.

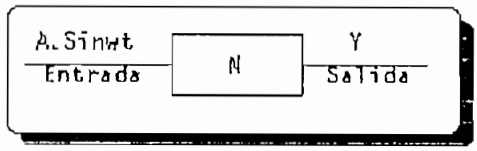



*Fig 1.3 Diagrama de bloques de un sistema no lineal*

$$Y=\sum_{n-1}^{\infty}[bn.Sin(nwt)+an.Cos(nwt)]$$

En donde:

$$a_n=\frac{2}{T}\int_0^T y(t)Cos(nwt)dt$$

$$b_n=\frac{2}{T}\int_0^T y(t)Sin(nwt)dt$$

Pero debido a que se pueden despreciar las armónicas superiores, ya que tal como se indicó generalmente las plantas son filtros pasabajos, y además la amplitud de las armónicas es siempre menor que la fundamental, entonces se tendrá:

$$Y(t)=b1Sinwt+a1Coswt$$

$$Y(t)=\sqrt{a1^2+b1^2}\left[\frac{b1}{\sqrt{a1^2+b1^2}}Sinwt+\frac{a1}{\sqrt{a1^2+b1^2}}Coswt\right]$$

Si se toma en cuenta que:

$$\frac{b1}{\sqrt{a1^2+b1^2}}=Cos\theta \quad y \quad \frac{a1}{\sqrt{a1^2+b1^2}}=Sin\theta$$

Entonces

$$Y(t)=\sqrt{a1^2+b1^2}\,Sin(wt+\theta)=BSin(wt+\theta)$$

$$B=\sqrt{a1^2+b1^2}$$

$$\theta=\tan^{-1}\frac{a1}{b1}$$

Quedando con ello la función descriptiva definida de la siguiente manera:

$$N(A,w) = \frac{B}{A}e^{j\theta} \qquad (1.3)$$

La ecuación anterior corresponde a la Función descriptiva de un sistema no lineal, la misma que depende tanto de la amplitud de la entrada A como de la frecuencia w.

La ecuación 1.3 puede ser normalizada si se la separa a esta en una parte real y una imaginaria, entonces

$$N(A,W) = g(A) + j\ b(A) \qquad (1.4)$$

y se puede llegar a demostrar fácilmente que:

$$g(A) = b1/A$$

$$b(A) = a1/A$$

Mediante este tipo de normalización indicado, se calculan a continuación para algunos tipos más comunes de no-linealidades, su función descriptiva.

## 1.2 EJEMPLOS DE DETERMINACION DE LA FUNCION DESCRIPTIVA EN ALGUNAS NO-LINEALIDADES CONOCIDAS

*1) Relé Ideal.-*

Este tipo de relé es conocido en control como ON-OFF puesto que presenta ya sea una conexión violenta o una desconexión similar. La representación de este relé se muestra en la figura 1.4.

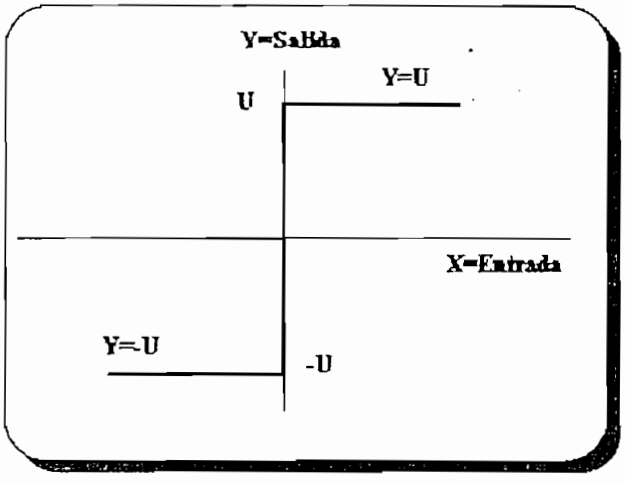



*Fig 1.4 Relé Ideal*

Primeramente como se observa de la figura 1.4, en el eje de las ordenadas está representada la salida, mientras que en el de las abcisas se encuentra la entrada, por lo tanto se debe determinar la ecuación correspondiente a la salida respecto a la entrada la misma que para este caso es:

$$Y=\begin{cases} U & A\geq 0 \\ -U & A<0 \end{cases} \qquad (1.5)$$

En donde Y representa la salida y A es la amplitud máxima de la señal de entrada X, ya que X es una señal de entrada sinusoidal de amplitud A

$$X=ASin(wt)$$

Entonces el diagrama correspondiente tanto a las señales de entrada como a la salida que presenta un relé ideal, aparece en la figura 1.5.

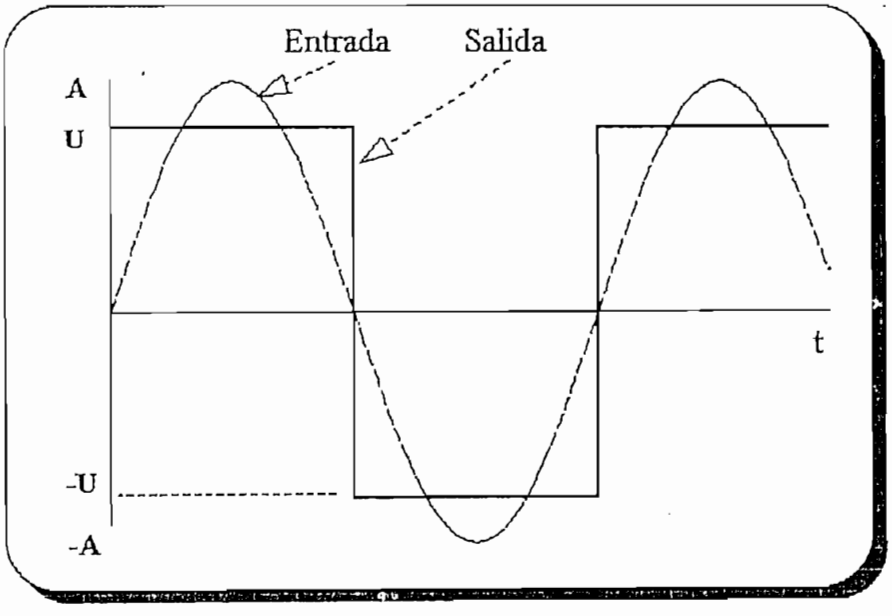



*Fig. 1.5 Respuesta de un relé ideal*

Considerando tanto la figura 1.5 como la ecuación 1.5 se procede a continuación a determinar la función descriptiva para este tipo de relé.

Como ya se indicó anteriormente, es necesario obtener la fundamental de la onda de salida, la misma que se la determina de la siguiente forma:

Se puede apreciar que la señal de salida es una onda cuadrada de período T, con paridad tipo impar, lo cual significa que la componente $a_1$ de la serie de Fourier es igual a cero, mientras que $b_1$ se la calcula así:

$$b_1 = \frac{2}{T}\int_0^T y(t)\,Sin(wt)dt = \frac{8U}{T}\int_0^{T/4} Sin(wt)dt$$

$$b_1 = -\frac{8U}{WT}\left(Cos\left(W\frac{T}{4}\right) - Cos\left(0\right)\right)$$

Tomando en cuenta que: $W = \frac{2\pi}{T}$

$$b_1 = \frac{4U}{\pi}$$

Se ha obtenido la fundamental de la serie de Fourier, quedando:

$$g(A) = \frac{4U}{\pi A}$$
$$b(A) = 0$$

así:

$$N = \frac{4U}{\pi A} \qquad (1.6)$$

Además, se aprecia en la figura 1.5 que la salida es una onda cuadrada cuyo cruce por cero se presenta exactamente al mismo tiempo que el cruce por cero de la señal de entrada sinusoidal, es por ello que en la ecuación 1.6 que corresponde a la función descriptiva no aparece retardo de fase entre la componente armónica fundamental de la salida y la onda sinusoidal de entrada; es decir $b(A)=0$.

En la ecuación 1.6 se ve que N solamente depende inversamente de "A" (amplitud de la entrada) y no de la frecuencia w. Esto se ilustra en la figura 1.6

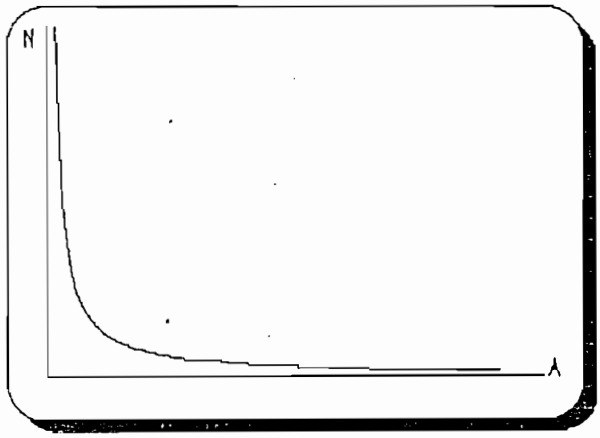



*Fig. 1.6 Respuesta de N en función de A para el caso de un relé ideal*

*2) Relé con zona muerta*

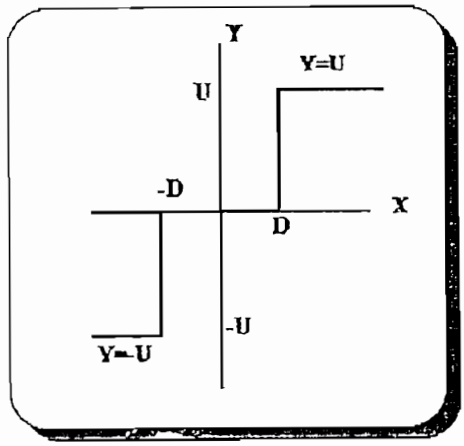



*Fig. 1.7 Relé con Zona Muerta*

Para este caso la ecuación que representa la salida respecto a la entrada es:

$$Y=\begin{cases} U & A\geq D \\ 0 & -D\leq A<D \\ -U & A<-D \end{cases} \qquad (I.7)$$

Donde X=Asin(wt)

Se puede ver de la característica del relé con zona muerta (figura 1.7), que pueden presentarse dos casos bien diferenciados, cuando la amplitud de la entrada es menor que D y cuando es mayor que D.

En el un caso cuando la entrada es menor que D la salida siempre es cero, por lo tanto la función descriptiva también será igual a cero, y no es necesario analizar en detalle este caso.

En el segundo caso en que A>D la salida corresponde a la figura 1.8, y se puede determinar la Función descriptiva de la siguiente forma:

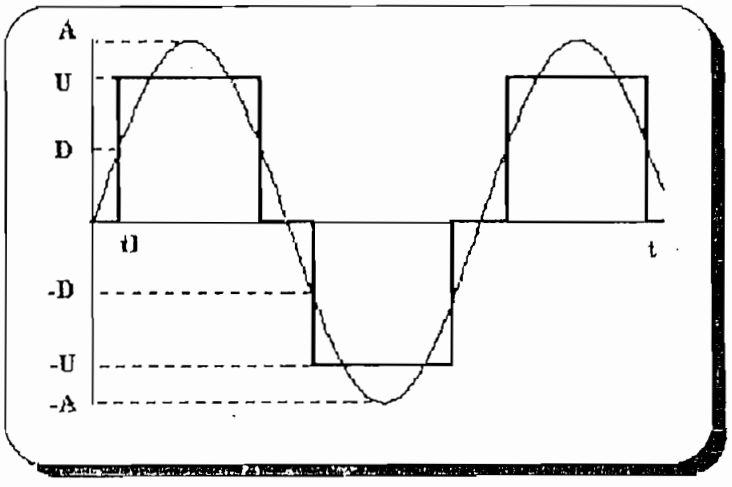



*Fig. 1.8 Respuesta de un relé con zona muerta*

Primero, $a_1 = 0$ por ser la salida una onda tipo impar, y con ello b(A)=0, mientras que $b_1$ se calcula:

$$D = A\mathrm{Sin}(wt_1)$$

$$\frac{D}{A} = \mathrm{Sin}(wt_1)$$

definiendo $wt_1=\theta_1$ entonces:

$$\theta_1=\mathrm{Sin}^{-1}\left(\frac{D}{A}\right)$$

$$b_1 = \frac{2}{T}\int_0^T y(t)\mathrm{Sin}(wt)dt = \frac{8U}{T}\int_{t1}^{T/4}\mathrm{Sin}(wt)dt = \frac{8U}{wT}\left(\mathrm{Cos}(wt_1)-\mathrm{Cos}(w\frac{T}{4})\right)$$

$$b_1 = \frac{4U}{\pi}\mathrm{Cos}(wt1) = \frac{4U}{\pi}\mathrm{Cos}\theta_1$$

entonces:

$$N=\begin{cases} \frac{4U}{\pi A}\mathrm{Cos}\theta_1 & A>D \\ 0 & A\leq D \end{cases} \qquad (1.8)$$

Si se grafica N en función de A se obtiene la figura 1.9.

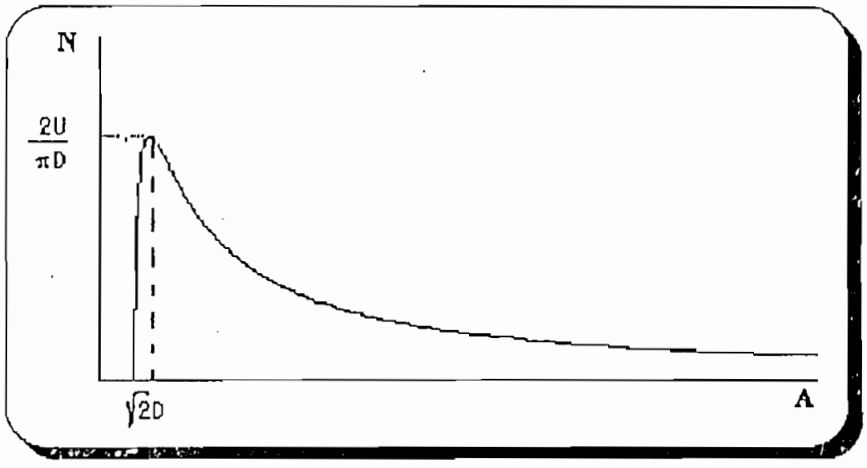



*Fig. 1.9 Gráfico de N(A) para el caso de un relé con zona muerta*

En el gráfico anterior se observa que existe un pico en la curva cuando la amplitud de entrada es $\sqrt{2}\,D$, siendo el pico de N igual a $\frac{2U}{\pi D}$, por lo tanto se puede notar que para el caso de un relé con zona muerta se

tendrá variaciones de la función descriptiva que no son directa ni indirectamente proporcionales a la variación de la amplitud de la señal de entrada "A".

*3) Relé con Pendiente.-*

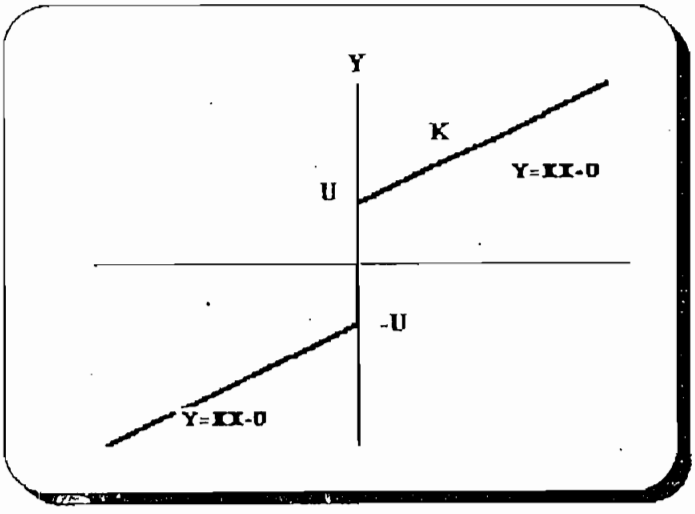



*Fig. 1.10 Relé con Pendiente*

La ecuación que representa la salida respecto a la entrada es:

$$Y=\begin{cases} KX+U & A\geq 0 \\ KX-U & A<0 \end{cases} \qquad (1.9)$$

donde X=ASin(wt), (ver figura 1.11)

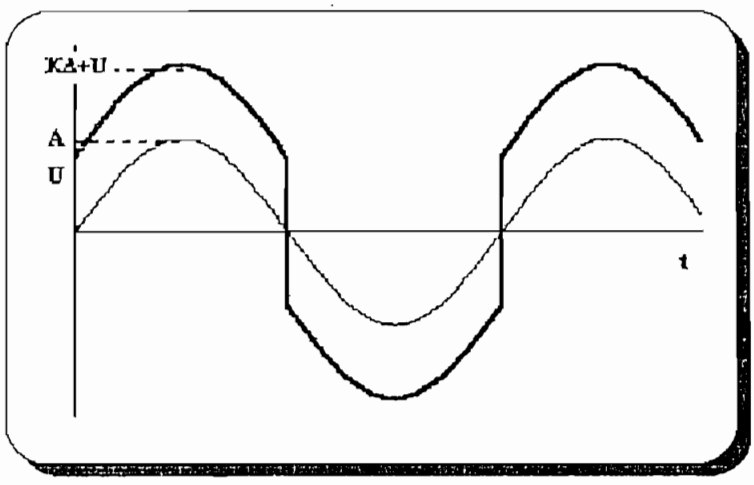



*Fig. 1.11 Respuesta de un Relé con Pendiente*

Como la salida es una onda con paridad impar entonces $a_1=0$, mientras que $b_1$ se obtiene de la siguiente manera:

$$b_1=\frac{8}{T}\int_0^{T/4}(KASinwt+U)*Sinwtdt = \frac{8}{T}\left[KA\int_0^{T/4}Sin^2(wt)dt+U\int_0^{T/4}Sin(wt)dt\right]$$

$$b_1=\frac{8}{T}\left[KA\int_0^{T/4}\frac{1-Cos(2wt)}{2}dt+\frac{U}{W}\Big(Cos0-Cos(wT/4)\Big)\right]$$

$$b_1=\frac{8}{T}\left[\frac{KA}{2}\left(t-\frac{Sin(2wt)}{2w}\right)_0^{T/4}+\frac{U}{W}(1+0)\right] = \frac{8}{2TW}\left[\frac{KA}{2}\left(2w\frac{T}{4}-Sin(2w\frac{T}{4})\right)+2U\right]$$

$$b_1=\frac{2}{\pi}\left(\frac{KA\pi}{2}+2U\right) = KA+\frac{4U}{\pi}$$

entonces:

$$N=K+\frac{4U}{\pi A} \qquad (1.10)$$

Se puede notar que la expresión anterior corresponde a la suma de una no linealidad cuya salida sea sinusoidal (N=K) más una cuya salida sea cuadrada ($N=4U/\pi A$).

*4) Relé con Zona Muerta y Pendiente.-*

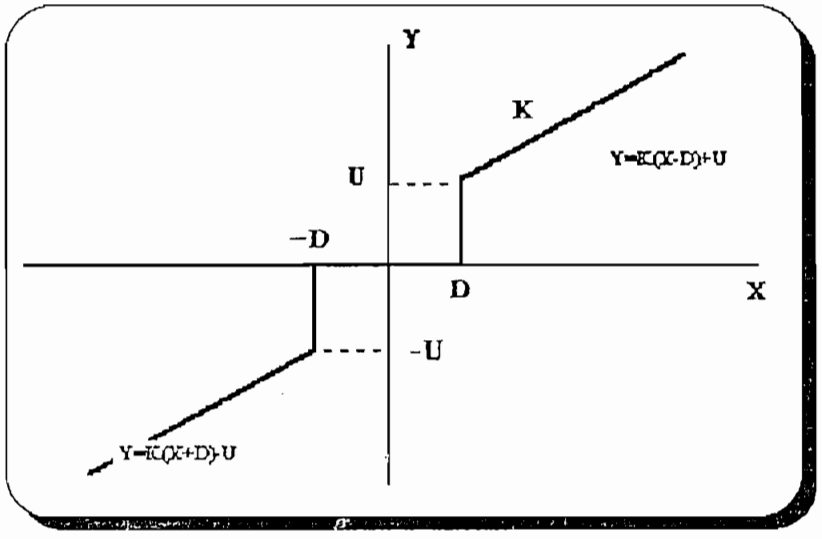



*Fig. 1.12 Relé con Zona Muerta y Pendiente*

Para este caso la ecuación que representa la salida respecto a la entrada es:

$$Y=\begin{cases} K(X-D)+U & A\geq D \\ 0 & -D\leq A<D \\ K(X+D)-U & A<-D \end{cases} \qquad (1.11)$$

donde X=ASin(wt)

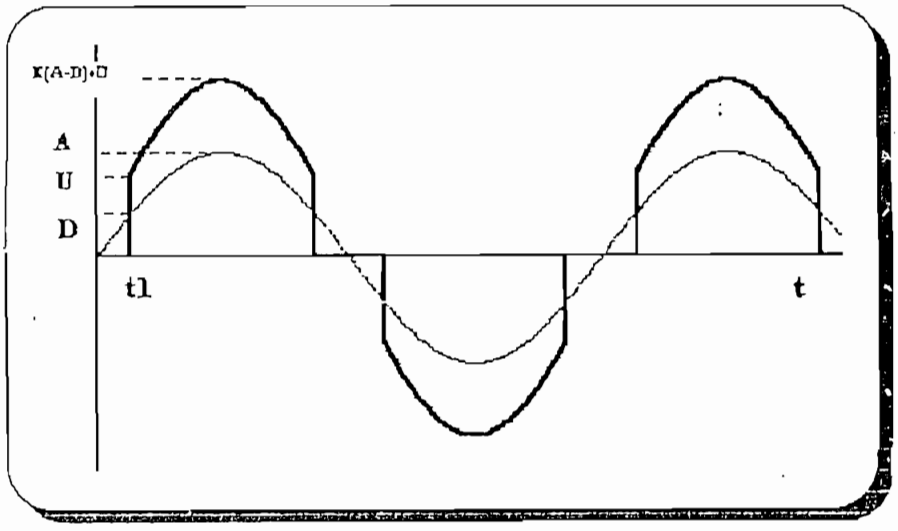



*Fig. 1.13 Respuesta de un Relé con Zona Muerta y Pendiente*

Entonces como la salida tiene paridad impar $a_1=0$ y $b_1$ se obtiene así:

$$b_1=\frac{8}{T}\int_{t_1}^{T/4}[K(ASinwt-D)+U]Sinwtdt = \frac{8}{T}\left[KA\int_{t_1}^{T/4}Sin^2wtdt+(U-KD)\int_{t_1}^{T/4}Sinwtdt\right]$$

$$b_1=\frac{8}{T}\left[KA\int_{t_1}^{T/4}\frac{1-Cos2wt}{2}dt+\frac{(U-KD)}{w}\left(Cos(wt_1)-Cos(w\frac{T}{4})\right)\right]$$

$$b_1=\frac{8}{2WT}\left[\frac{KA}{2}\left(\pi-2\theta_1-Sin\pi+Sin(2\theta_1)\right)+2(U-KD)Cos(\theta_1)\right]$$

$$b_1 = \frac{KA}{\pi}\left(\pi-2\theta_1+Sin(2\theta_1)\right)+\frac{4}{\pi}(U-KD)Cos(\theta_1)$$

entonces:

$$N = \frac{K}{\pi}\left(\pi-2\theta_1+Sin(2\theta_1)\right)+\frac{4}{\pi A}(U-KD)Cos(\theta_1); \qquad A>D \qquad (1.12)$$

La respuesta anterior tal como se indica solo es válida si A>D ya que en caso contrario la salida siempre valdrá 0 y con ello N=0, por lo tanto no interesa analizar con detenimiento dicho caso.

*5) Saturación Ideal.-*

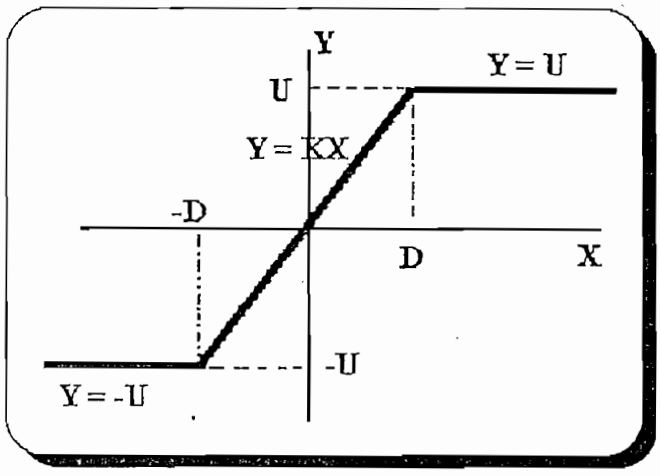



*Fig. 1.14 Saturación Ideal*

Según la figura 1.14, la característica es:

$$Y = \begin{cases} U & A \geq D \\ KX & -D \leq A < D \\ -U & A < -D \end{cases} \qquad (1.13)$$

donde X = ASin(wt)

Para este tipo de no linealidad se tendrán dos casos bien diferenciados, uno en el cual la amplitud de la entrada no sobrepase el valor de D y otro en que si lo haga.

Para el primer caso lo que se obtendría es una amplificación de la señal de entrada con una ganancia de K = U/D, por lo tanto la señal de salida también será una sinusoide quedando definida la función descriptiva como la división de la amplitud de la salida respecto a la entrada de la siguiente manera:

$$N=\frac{kA}{A}=k=\frac{U}{D} \qquad (1.14)$$

La ecuación anterior corresponde al caso en el que $A \leq D$

En el segundo caso, cuando la entrada sobrepasa el valor de D, entonces la salida tendrá un recorte en su señal, tal como se muestra en la figura 1.15

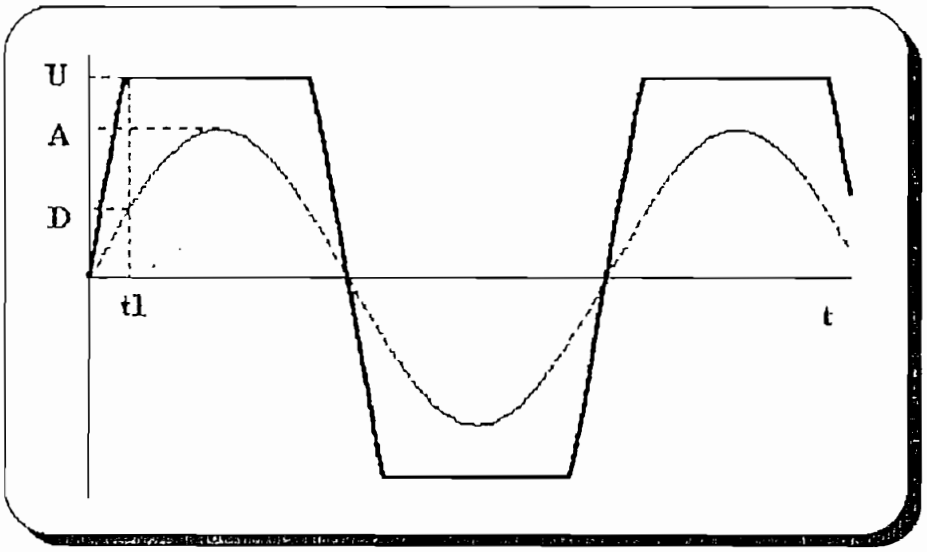



*Fig. 1.15 Respuesta que presenta una Saturación Ideal, para A>D*

Siguiendo el procedimiento de los casos anteriores:

$a_1=0$ (Paridad Impar)

$$b_1=\frac{8}{T}\left[\int_0^{t_1} KASin^2wtdt+\int_{t_1}^{T/4} USinwtdt\right] = \frac{8}{2WT}\left[\frac{KA}{2}\left(2wt_1-Sin(2wt_1)\right)+2UCos(\theta_1)\right]$$

$$b_1=\frac{KA}{\pi}\left(2\theta_1-Sin(2\theta_1)\right)+\frac{4U}{\pi}Cos(\theta_1)$$

$$N=\frac{K}{\pi}\left(2\theta_1-Sin(2\theta_1)\right)+\frac{4U}{\pi A}Cos(\theta_1) \qquad (1.15)$$

En donde

$$wt_1 = Sin^{-1}\frac{D}{A} = \theta_1$$

Como ya se indicó, la expresión anterior sólo es útil para cuando A>D.

*6) Saturación con Zona Muerta.-*

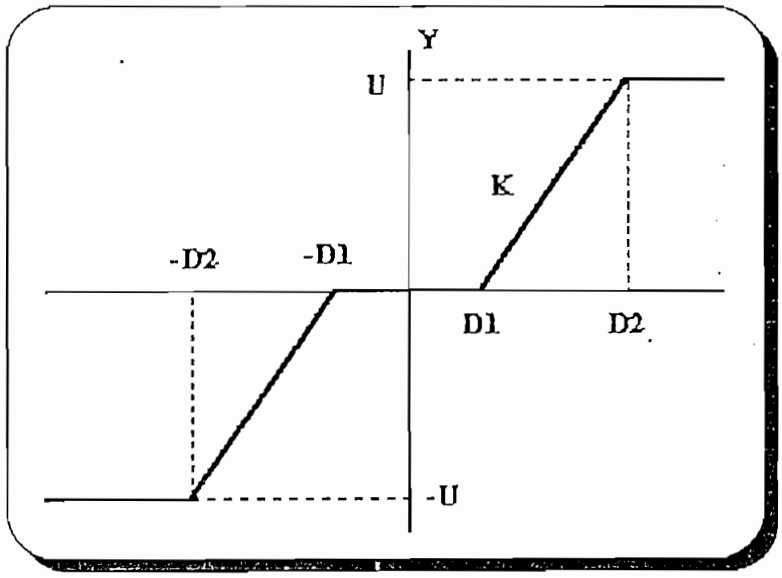



*Fig. 1.16 Saturación con Zona Muerta*

La salida en función de la entrada queda:

$$Y = \begin{cases} U & A \geq D_2 \\ K(X-D_1) & D_1 \leq A < D_2 \\ 0 & -D_1 \leq A < D_1 \\ K(X+D_1) & -D_2 \leq A < -D_1 \\ -U & A < -D_2 \end{cases} \qquad (1.16)$$

donde X = ASin(wt)

En este tipo de alinealidad se aprecia que cuando $-D_1 \leq A < D_1$ la salida es cero, por ello N=0; entonces, en esta no linealidad se pueden presentar dos casos importantes ya que la amplitud de la señal de entrada puede ser menor que $D_2$ pero mayor o igual a $D_1$ en cuyo caso la salida es de un tipo diferente al que se obtendría si la entrada es mayor que $D_2$. Es por ello que a continuación se analizan los dos casos.

*a)* $D_1 \leq A < D_2$

Para este caso la salida que se presenta es la siguiente.

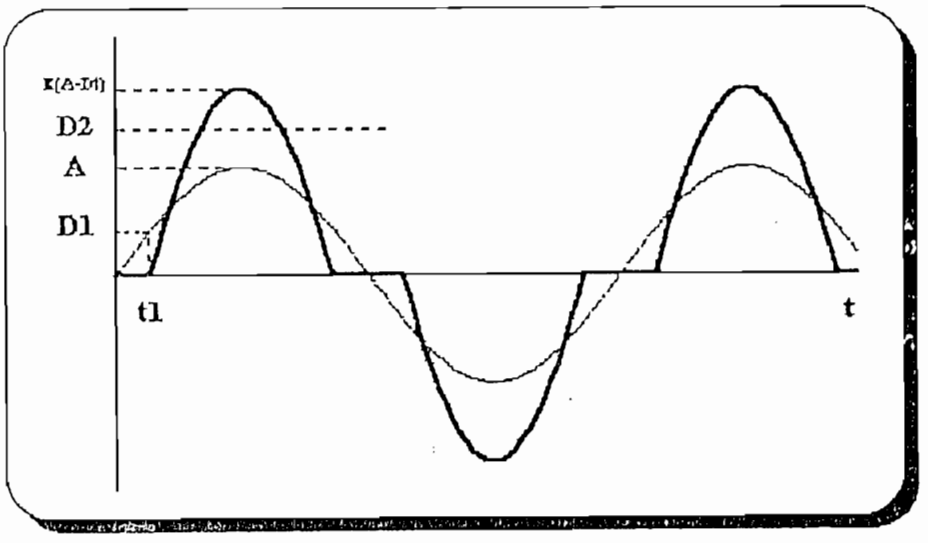



*Fig. 1.17 Respuesta de una Saturación con Zona muerta, con* $D_1 \leq A < D_2$

Como puede verse se trata de una onda con paridad tipo impar por lo tanto $a_1=0$, mientras que $b_1$ resulta:

$$b_1=\frac{8}{T}\int_{t_1}^{T/4} K(A\,Sinwt-D_1)Sinwtdt = \frac{8K}{T}\left[A\int_{t_1}^{T/4} Sin^2wtdt-D_1\int_{t_1}^{T/4} Sinwtdt\right]$$

$$b_1=\frac{8K}{T}\left[A\int_{t_1}^{T/4}\frac{1-Cos2wt}{2}dt+\frac{D_1}{W}\Big(0-Cos(wt_1)\Big)\right] = \frac{8K}{2WT}\left[\frac{A}{2}\Big(\pi-2wt_1+Sin(2wt_1)\Big)-2D_1Cos(wt_1)\right]$$

$$b_1=\frac{AK}{\pi}\left(\pi-2\theta_1+\mathrm{Sin}(2\theta_1)\right)-\frac{4KD1}{\pi}\mathrm{Cos}\theta_1$$

$$N=\frac{K}{\pi}\left(\pi-2\theta_1+\mathrm{Sin}(2\theta_1)\right)-\frac{4K}{\pi A}D_1\mathrm{Cos}\theta_1\ ;\ D_1\leq A<D_2 \qquad (1.17)$$

En donde:

$$wt_1=\theta_1; \qquad y \qquad A\mathrm{Sin}(wt_1)=D_1$$

$$\theta_1=\mathrm{Sin}^{-1}(D_1/A)$$

*b) $A>D_2$*

En este segundo caso que se presenta para este tipo de no linealidad se aprecia que la salida para una entrada sinusoidal es como la indicada en la figura 1.18.

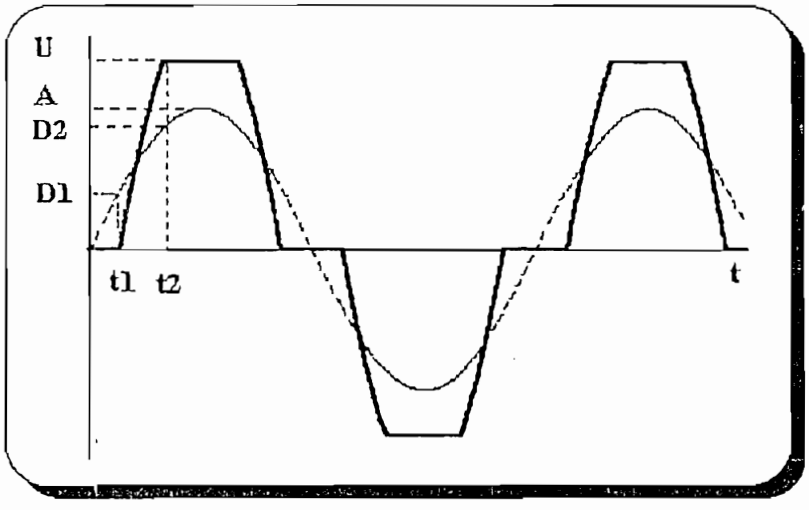



*Fig. 1.18 Respuesta de una Saturación con Zona Muerta, $A>D_2$*

También para este tipo de onda se tiene que $a_1=0$ por tratarse de una onda con paridad impar, pero $b_1$ cambia siendo en este caso:

$$b_1=\frac{8}{T}\left[\int_{t_1}^{t_2} K(A\,Sinwt-D_1)Sinwtdt+\int_{t_2}^{T/4} USinwtdt\right]$$

$$b_2=\frac{8}{T}\left[KA\int_{t_1}^{t_2}\frac{1-Coswt}{2}dt-KD_1\int_{t_1}^{t_2}Sinwtdt+U\int_{t_2}^{T/4}Sinwtdt\right]$$

$$b_1=\frac{KA}{\pi}\Big(2\theta_2-2\theta_1-Sin(2\theta_2)+Sin(2\theta_1)\Big)+\frac{4K}{\pi}D_1\Big(Cos(\theta_2)-Cos(\theta_1)\Big)+\frac{4}{\pi}UCos(\theta_2)$$

$$N=\frac{K}{\pi}\Big(2(\theta_2-\theta_1)-Sin(2\theta_2)+Sin(2\theta_1)\Big)+\frac{4K}{\pi A}D_1\Big(Cos(\theta_2)-Cos(\theta_1)\Big)+\frac{4}{\pi A}UCos(\theta_2) \qquad (1.18)$$

en donde

$$wt_1=\theta_1 \quad y \quad wt_2=\theta_2$$
$$ASinwt_1=D_1 \quad y \quad ASinwt_2=D_2$$
$$\theta_1=Sin^{-1}D_1/A \quad y \quad \theta_2=Sin^{-1}D_2/A$$

*7) Saturación con Pendiente.-*

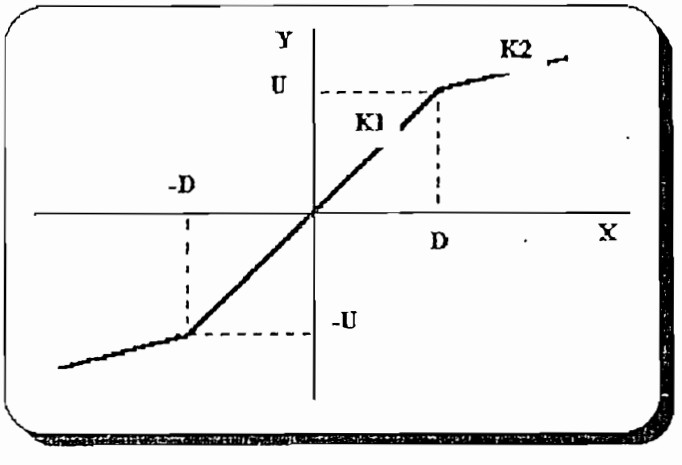



*Fig. 1.19 Saturación con pendiente*

La ecuación de la entrada respecto a la salida es:

$$Y=\begin{cases} K_2(X-D)+U & A\geq D \\ K_1X & -D\leq A<D \\ K_2(X+D)-U & A<-D \end{cases} \qquad (1.19)$$

donde X = ASin(wt)

Para esta no linealidad la forma de onda de la salida es:

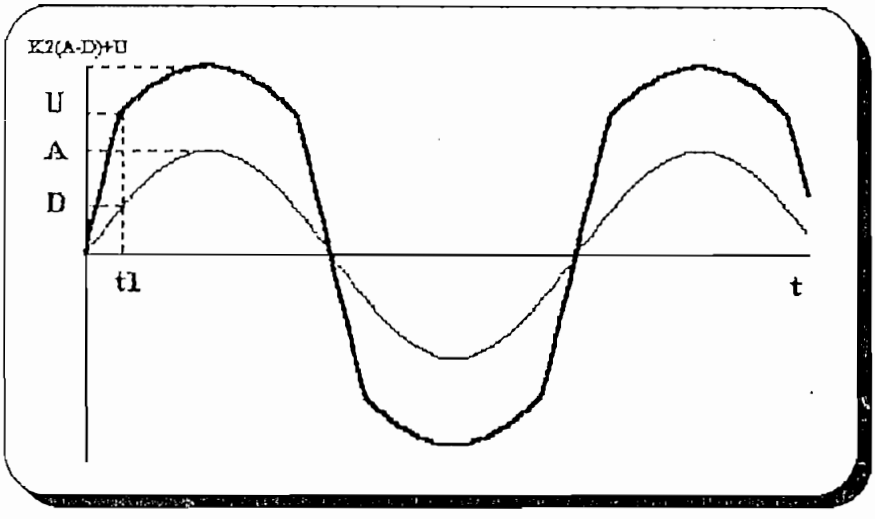



*Fig. 1.20 Respuesta de una saturación con pendiente*

$$a_1=0$$

$$b_1=\frac{8}{T}\left[\int_0^{t_1} K_1ASin^2wtdt+\int_{t_1}^{T/4}\left(K_2(ASinwt-D)+U\right)Sinwtdt\right]$$

$$b_1=\frac{8}{2wT}\left[\frac{K_1A}{2}\left(2wt_1-Sin(2wt_1)\right)+\frac{K_2A}{2}\left(2w\frac{T}{4}-2wt_1-Sin(2\frac{WT}{4})+Sin(2wt_1)\right)+2(U-K_2D)Coswt_1\right]$$

$$b_1=\frac{K_1A}{\pi}\left(2\theta_1-Sin(2\theta_1)\right)+\frac{K_2A}{\pi}\left(\pi-2\theta_1+Sin(2\theta_1)\right)+\frac{4U}{\pi}Cos\theta_1-\frac{4}{\pi}K_2DCos\theta_1$$

$$N=\frac{K_1}{\pi}\left(2\theta_1-Sin(2\theta_1)\right)+\frac{K_2}{\pi}\left(\pi-2\theta_1+Sin(2\theta_1)\right)-\frac{4}{\pi A}K_2DCos\theta_1+\frac{4U}{\pi A}Cos\theta_1 \qquad (1.20)$$

*8) Saturación con Zona Muerta y Pendiente*

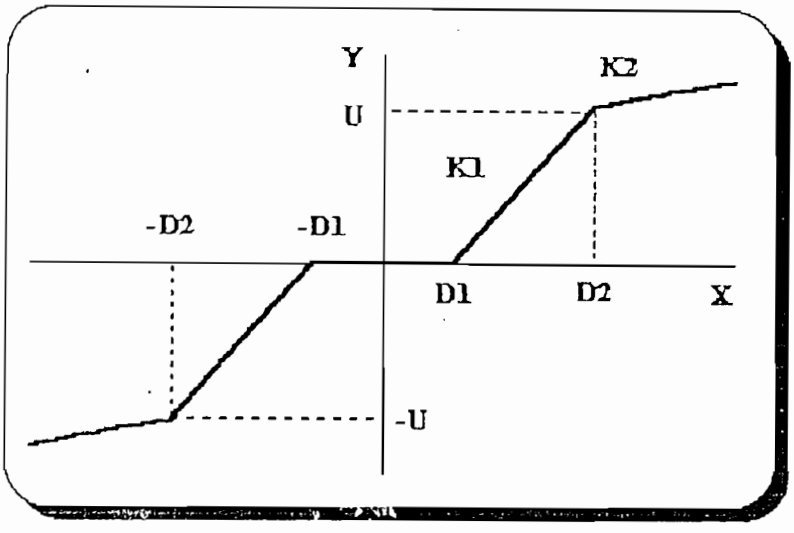



*Fig. 1.21 Saturación con Zona Muerta y Pendiente*

$$Y=\begin{cases} K_2(X-D_2)+U & A\geq D_2 \\ K_1(X-D_1) & D_1\leq A<D_2 \\ 0 & -D_1\leq A<D_1 \\ K_1(X+D_1) & -D_2\leq A<-D_1 \\ K_2(X+D_2)-U & A<-D_2 \end{cases} \qquad (1.21)$$

donde X = ASin(wt)

Si la amplitud de la entrada resulta ser menor que $D_2$ entonces nos encontramos frente al mismo caso (a) analizado en la no linealidad 6 (Saturación con Zona muerta).

La salida que se presenta con esta no linealidad cuando la amplitud de la señal de entrada es superior a $D_2$ es como la indicada en la figura 1.22

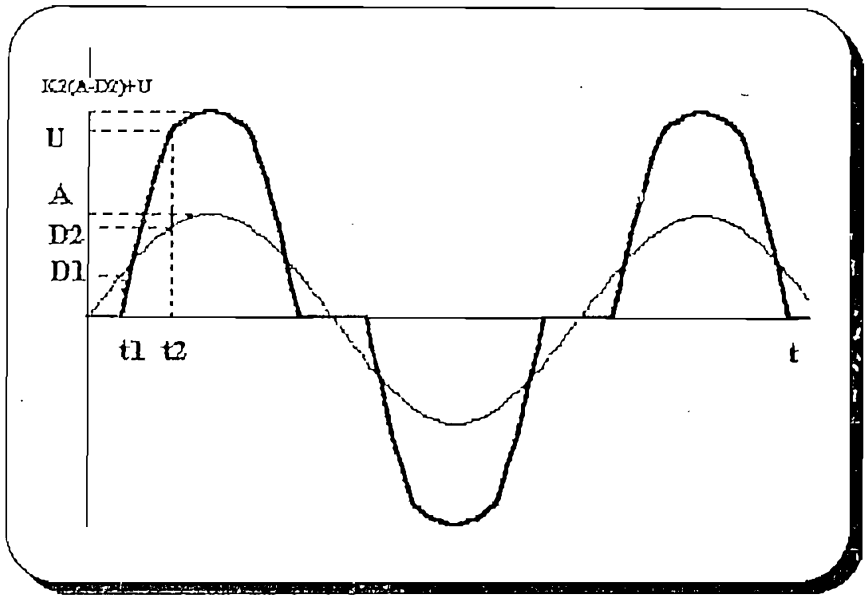



*Fig. 1.22 Respuesta de una NL con Zona Muerta y Pendiente*

Para este caso se tiene que:

$$a_1=0$$

$$b_1=\frac{8}{T}\left[\int_{t_1}^{t_2} K_1(A\,Sinwt-D_1)Sinwtdt+\int_{t_2}^{T/4}\left(K_2(A\,Sinwt-D_2)+U\right)Sinwtdt\right]$$

$$b_1=\frac{8}{T}\left[K_1A\int_{t_1}^{t_2}\frac{1-Cos2wt}{2}dt-K_1D_1\int_{t_1}^{t_2}Sinwtdt+K_2A\int_{t_2}^{T/4}\frac{1-Cos2wt}{2}dt+(U-K_2D_2)\int_{t_2}^{T/4}Sinwtdt\right]$$

resolviendo estas integrales se llega a:

$$b_1=\frac{K_1A}{\pi}\left(2\theta_2-2\theta_1-Sin2\theta_2+Sin2\theta_1\right)+\frac{4K_1D_1}{\pi}\left(Cos\theta_2-Cos\theta_1\right)+$$
$$+\frac{K_2A}{\pi}\left(\pi-2\theta_2+Sin2\theta_2\right)+\frac{4}{\pi}(U-K_2D_2)Cos\theta_2$$

$$N=\frac{K_1}{\pi}\left(2\theta_2-2\theta_1-Sin2\theta_2+Sin2\theta_1\right)+\frac{4K_1D_1}{\pi A}\left(Cos\theta_2-Cos\theta_1\right)+$$
$$+\frac{K_2}{\pi}\left(\pi-2\theta_2+Sin2\theta_2\right)-\frac{4K_2D_2}{\pi A}Cos\theta_2+\frac{4U}{\pi A}Cos\theta_2 \qquad (1.22)$$

*9) Histéresis Ideal.-*

El presente caso es muy común dentro de los procesos industriales, y el diagrama correspondiente a este tipo de alinealidad se muestra en la figura 1.23

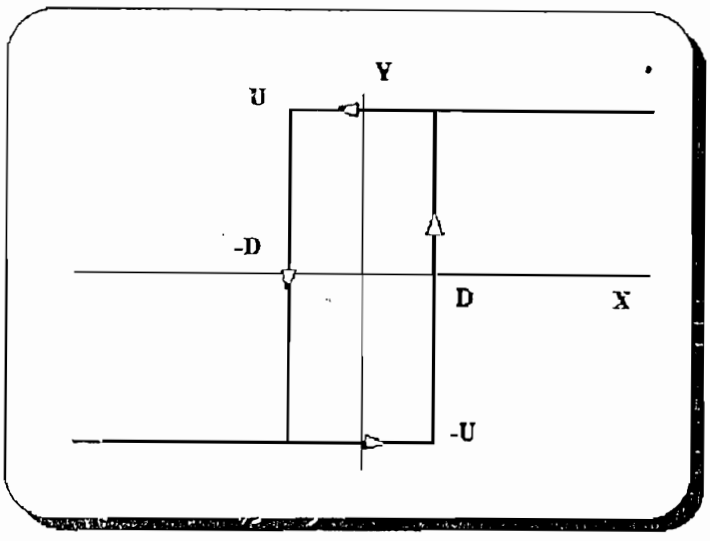



*Fig. 1.23 Histéresis Ideal*

Asumiendo primeramente que la salida se encuentra en el valor -U, y la entrada esta en el semiciclo negativo, conforme se mueve la señal de entrada llega un momento en el cual pasa al semiciclo positivo, y conforme la amplitud de la entrada va creciendo, llega un momento en el cual sobrepasa al valor +D, justo en ese instante la salida toma el valor de +U y se mantiene en ese punto hasta que la señal de entrada pase al semiciclo negativo y rebase el valor de -D en cuyo instante la salida toma el valor de -U y se mantiene así repitiendo nuevamente el ciclo, esto se observa con claridad en la figura 1.23 (sentido de la flecha)

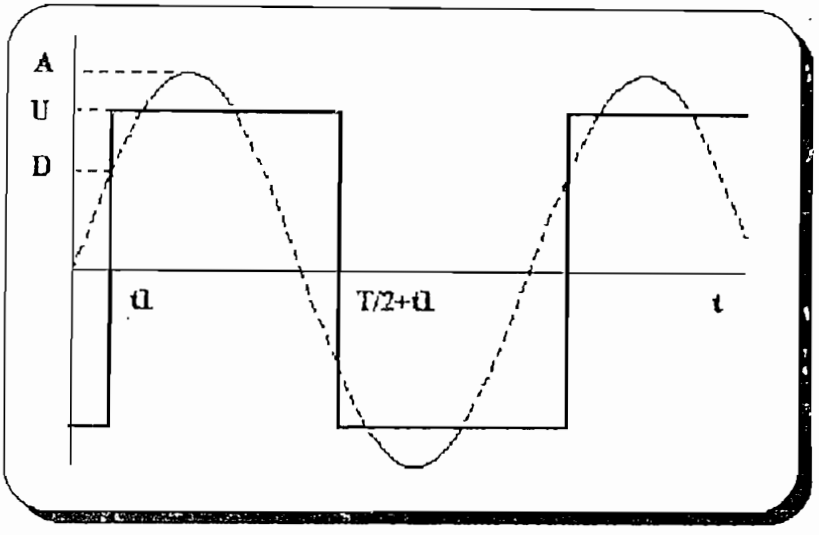



*Fig. 1.24 Respuesta de una Histéresis Ideal*

Se puede apreciar que la respuesta del relé con histéresis resulta similar a la del relé común, pero presenta un defasamiento de $t_1$, es decir la salida se atrasa $t_1$ respecto a la entrada, por esta razón la componente fundamental de la serie de Fourier será la misma si se cambia de referencia a $t_1$, de tal manera que:

$$b_1 = \frac{4U}{\pi}$$

$a_1 = 0$ (por ser función impar)

definiendo $wt_1 = \theta$

$$A\text{Sin}(wt_1) = D$$

$$wt_1 = \text{Sin}^{-1}(D/A)$$

$$\theta = \text{Sin}^{-1}\frac{D}{A}$$

$$N = \frac{4U}{\pi A}e^{-j\theta} = \frac{4U}{\pi A}(\text{Cos}\theta - j\text{Sin}\theta)$$

$$N = \begin{cases} \frac{4U}{\pi A}\text{Cos}\theta & \text{(Real)} \\ -\frac{4U}{\pi A}\text{Sin}\theta & \text{(Imaginaria)} \end{cases} \qquad (1.23)$$

En esta expresión aparecen dos términos, que como puede verse, el primero corresponde a la parte real de la función descriptiva, mientras que el segundo representa la parte imaginaria de dicha función, que es la forma como se va ha representar la función descriptiva de aquí en adelante.

Hay que destacar que la expresión 1.23 es válida únicamente cuando la entrada A sea mayor que D. Además se observa que la función descriptiva en este caso tiene una componente real y una imaginaria, cosa que no se presentaba en los casos anteriores en donde la señal de salida presenta paridad impar.

*10) Histéresis con Zona Muerta*

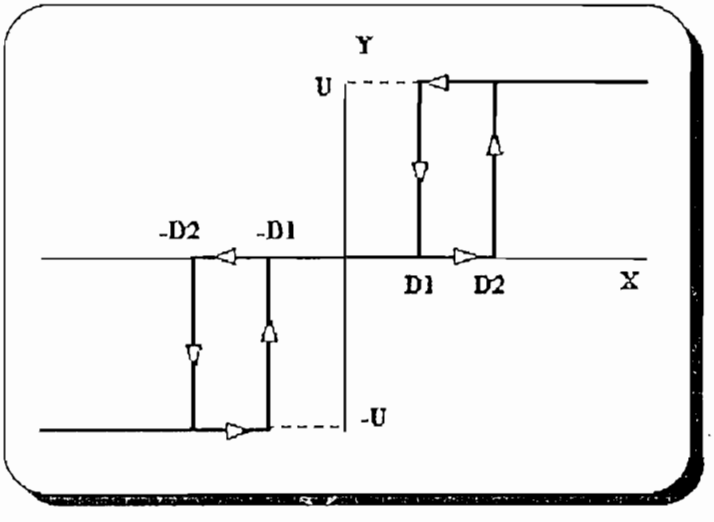



*Fig. 1.25 Histéresis con Zona Muerta*

Este tipo de no linealidad presenta una zona muerta de ancho igual $2D_2$ y una histéresis de ancho $(D_2-D_1)$. El comportamiento que se observa es de la siguiente manera; Mientras que la amplitud de la entrada en el

semiciclo positivo se mantenga por debajo del valor de $D_2$ la salida permanecerá en cero, es decir el elemento alineal permanecerá inactivo, luego al sobrepasar el valor de $D_2$ la salida toma inmediatamente el valor de +U y se mantiene en dicho valor hasta que la señal de entrada baje del punto $D_1$ entonces la salida se pone en cero. Cosa similar pasa en el semiciclo negativo de la señal de entrada, teniéndose como resultado la figura 1.26

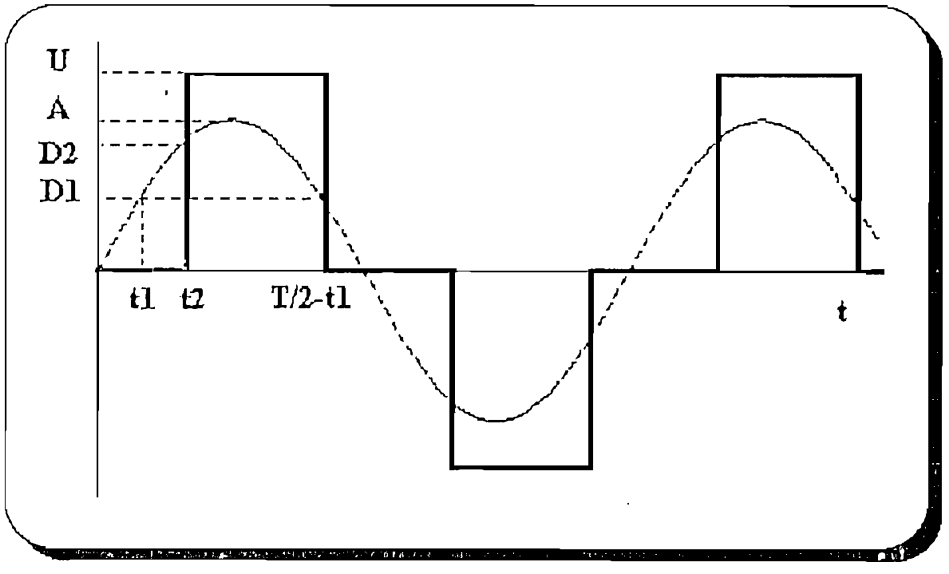



*Fig. 1.26 Respuesta de una Histéresis con Zona Muerta*

Como se observa de la figura anterior esta señal no posee paridad par como tampoco impar, por lo tanto se tendrá las dos componentes fundamentales de Fourier $a_1$ y $b_1$, las mismas que se las obtiene de la siguiente manera.

$$a_1=\frac{4}{T}\int_{t_2}^{\frac{T}{2}-t_1} UCoswtdt = \frac{4U}{Tw}\left(Sin(w\frac{T}{2}-wt_1)-Sin(wt_2)\right) = \frac{2U}{\pi}\left(Sin(\pi-wt_1)-Sin(wt_2)\right)$$

$$a_1=\frac{2U}{\pi}\left(Sin\theta_1-Sin\theta_2\right)$$

$$b_1=\frac{4}{T}\int_{t_2}^{\frac{T}{2}-t_1} USinwtdt = \frac{4U}{Tw}\left(Cos(wt_2)-Cos(w\frac{T}{2}-wt_1)\right) = \frac{2U}{\pi}\left(Cos(wt_2)-Cos(\pi-wt_1)\right)$$

$$b_1=\frac{2U}{\pi}\left(Cos\theta_2+Cos\theta_1\right)$$

$$N=\begin{cases}\frac{2U}{\pi A}\left(Cos\theta_1+Cos\theta_2\right) & \text{(Real)}\\ \frac{2U}{\pi A}\left(Sin\theta_1-Sin\theta_2\right) & \text{(Imaginaria)}\end{cases} \qquad (1.24)$$

en donde

$$wt_1=\theta_1 \quad y \quad wt_2=\theta_2$$
$$ASin\theta_1=D_1 \quad y \quad ASin\theta_2=D_2$$
$$\theta_1=Sin^{-1}D_1/A \quad y \quad \theta_2=Sin^{-1}D_2/A$$

*11) Histéresis con Pendiente*

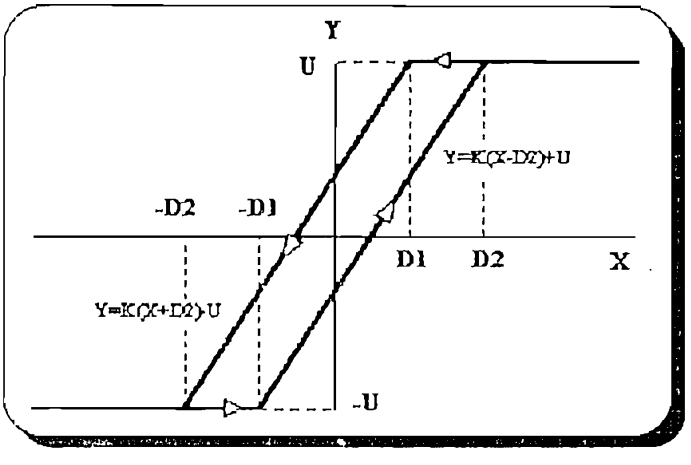



*Fig. 1.27 Histéresis con Pendiente*

$$K=\frac{2U}{D_2+D_1}$$

Para esta no-linealidad se obtiene la siguiente señal de salida

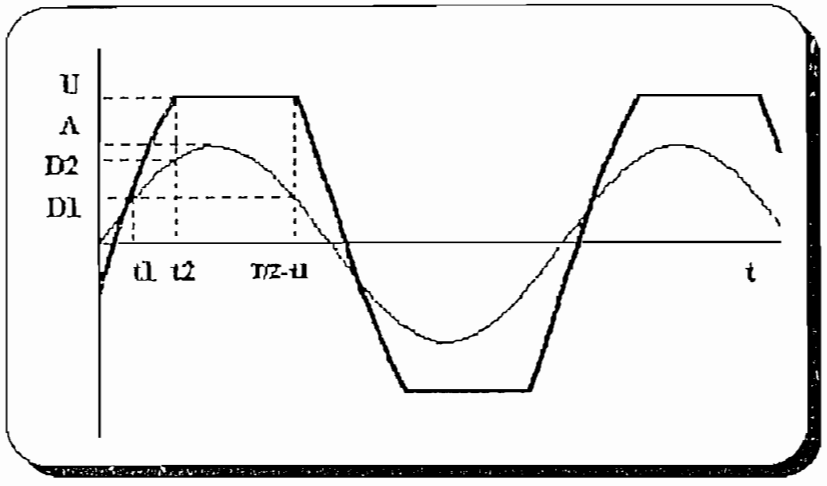



*Fig. 1.28 Respuesta de una Histéresis con Pendiente*

Para este caso el cálculo de la función descriptiva puede obtenerse de la siguiente manera:

$$a_1=\frac{4}{T}\left[\int_0^{t_2}\Big(K(A\,Sinwt-D_2)+U\Big)Coswtdt+\int_{t_2}^{T/2-t_1}U\,Coswtdt+\int_{T/2-t_1}^{T/2}\Big(K(A\,Sinwt+D_2)-U\Big)Coswtdt\right]$$

$$a_1=\frac{4}{T}\left[\frac{KA}{2}\frac{(Cos0-Cos2wt_2)}{2w}+\frac{(U-KD_2)}{w}Sinwt_2+\frac{U}{w}(Sinwt_1-Sinwt_2)\right]+$$
$$+\frac{4}{T}\left[\frac{KA}{2}\frac{Cos(2wT/2-2wt_1)-Cos(2wT/2)}{2w}+\frac{(U-KD_2)}{w}Sinwt_1\right]$$

$$a_1=\frac{KA}{2\pi}(Cos2\theta_1-Cos2\theta_2)+\frac{2}{\pi}(U-KD_2)(Sin\theta_2+Sin\theta_1)+\frac{2U}{\pi}(Sin\theta_1-Sin\theta_2)$$

$$b_1=\frac{4}{T}\left[\int_0^{t_2}\Big(K(A\,Sinwt-D_2)+U\Big)Sinwtdt+\int_{t_2}^{T/2-t_1}U\,Sinwtdt+\int_{T/2-t_1}^{T/2}\Big(K(A\,Sinwt+D_2)-U\Big)Sinwtdt\right]$$

$$b_1=\frac{4}{2wT}\left[\frac{KA}{2}(2wt_2-Sin2wt_2)+2(U-KD_2)(1-Coswt_2)+2U(Coswt_2+Coswt_1)\right]+$$
$$+\frac{4}{2wT}\left[\frac{KA}{2}\Big(2wt_1+Sin(2wT/2-2wt_1)\Big)-2(U-KD_2)(1-Coswt_1)\right]$$

$$b_1=\frac{KA}{2\pi}(2\theta_1+2\theta_2-Sin2\theta_1-Sin2\theta_2)+2\frac{(U-KD_2)}{\pi}(Cos\theta_1-Cos\theta_2)+2\frac{U}{\pi}(Cos\theta_2+Cos\theta_1)$$

$$N=\begin{cases} \frac{K}{2\pi}(2\theta_1+2\theta_2-Sin2\theta_1-Sin2\theta_2)+2\frac{(U-KD_2)}{\pi A}(Cos\theta_1-Cos\theta_2)+\frac{2U}{\pi A}(Cos\theta_2+Cos\theta_1) \\ \frac{K}{2\pi}(Cos2\theta_1-Cos2\theta_2)+\frac{2}{\pi A}(U-KD_2)(Sin\theta_2+Sin\theta_1)+\frac{2U}{\pi A}(Sin\theta_1-Sin\theta_2) \end{cases} \quad (1.25)$$

$$\begin{aligned} wt_1=\theta_1 \quad & y \quad wt_2=\theta_2 \\ ASin(wt_1)=D_1 \quad & y \quad ASin(wt_2)=D_2 \\ \theta_1=Sin^{-1}D_1/A \quad & y \quad \theta_2=Sin^{-1}D_2/A \end{aligned}$$

*12) Histéresis con Zona Muerta y Pendiente.-*

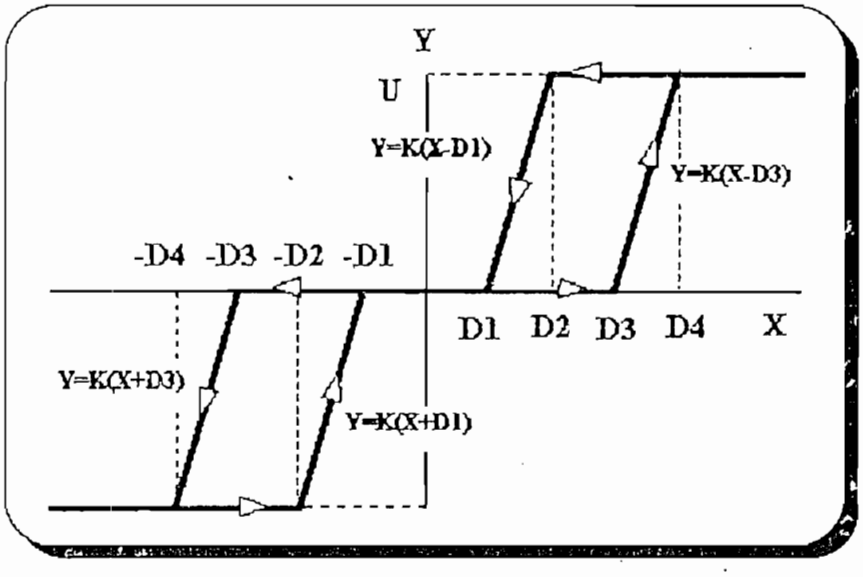



*Fig. 1.29 Histéresis con Zona Muerta y Pendiente*

$$K=\frac{U}{D_4-D_3}=\frac{U}{D_2-D_1}$$

Para este caso la señal de salida es como la indicada en la figura 1.30

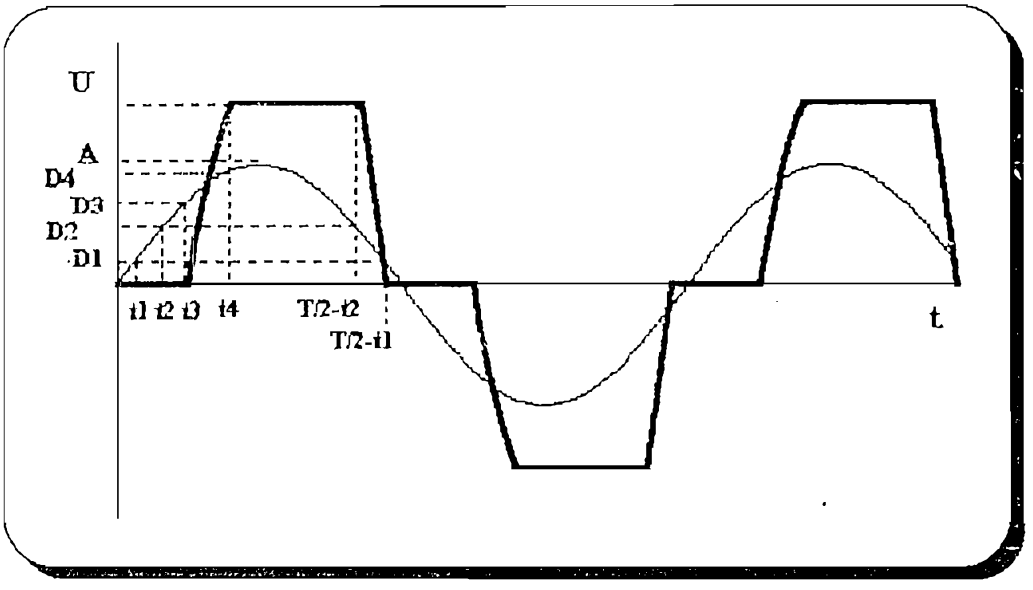



*Fig. 1.30 Respuesta de una Histéresis con Zona Muerta y Pendiente*

$$a_1=\frac{4}{T}\left[\int_{t_3}^{t_4} K(A\,Sinwt-D_3)Coswtdt+\int_{t_4}^{T/2-t_2} UCoswtdt+\int_{T/2-t_2}^{T/2-t_1} K(A\,Sinwt-D_1)Coswtdt\right]$$

$$a_1=\frac{KA}{2\pi}(Cos2\theta_2-Cos2\theta_1+Cos2\theta_3-Cos2\theta_4)-\frac{2KD_3}{\pi}(Sin\theta_4-Sin\theta_3)-$$
$$-\frac{2KD_1}{\pi}(Sin\theta_1-Sin\theta_2)+\frac{2U}{\pi}(Sin\theta_2-Sin\theta_4)$$

$$b_1=\frac{4}{T}\left[\int_{t_3}^{t_4} K(A\,Sinwt-D_3)Sinwtdt+\int_{t_4}^{T/2-t_2} USinwtdt+\int_{T/2-t_2}^{T/2-t_1} K(A\,Sinwt-D_1)Sinwtdt\right]$$

$$b_1=\frac{KA}{2\pi}\left(2(\theta_2-\theta_1+\theta_4-\theta_3)+Sin2\theta_1-Sin2\theta_2+Sin2\theta_3-Sin2\theta_4\right)+\frac{2KD_3}{\pi}(Cos\theta_4-Cos\theta_3)+$$
$$+\frac{2KD_1}{\pi}(Cos\theta_2-Cos\theta_1)+\frac{2U}{\pi}(Cos\theta_4+Cos\theta_2)$$

$$N=\begin{cases}\frac{K}{2\pi}\left(2(\theta_2-\theta_1+\theta_4-\theta_3)+Sin2\theta_1-Sin2\theta_2+Sin2\theta_3-Sin2\theta_4\right)+\frac{2KD_3}{\pi A}(Cos\theta_4-Cos\theta_3)+\\ +\frac{2KD_1}{\pi A}(Cos\theta_2-Cos\theta_1)+\frac{2U}{\pi A}(Cos\theta_4+Cos\theta_2)\\ \frac{K}{2\pi}(Cos2\theta_2-Cos2\theta_1+Cos2\theta_3-Cos2\theta_4)-\frac{2KD_3}{\pi A}(Sin\theta_4-Sin\theta_3)-\\ -\frac{2KD_1}{\pi A}(Sin\theta_1-Sin\theta_2)+\frac{2U}{\pi A}(Sin\theta_2-Sin\theta_4)\end{cases}$$

(1.26)

## 1.3 CRITERIO DE NYQUIST (ESTABILIDAD)

La función de transferencia de un sistema en lazo cerrado como el de la figura 1.31 puede expresarse de la siguiente manera

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)H(s)} \qquad (1.27)$$

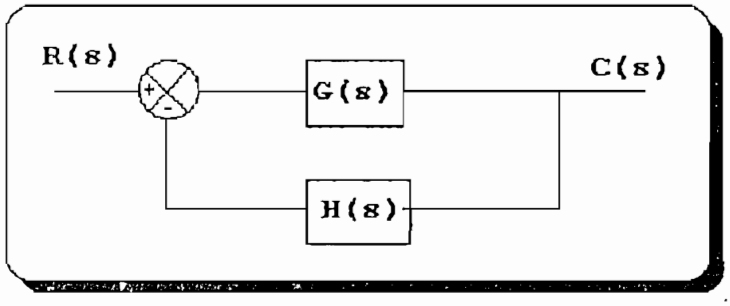



*Fig. 1.31 Sistema en lazo cerrado*

Analizando la función de transferencia C(s)/R(s), se tendrá estabilidad cuando ella no posea polos en el semiplano derecho del plano s, o sobre el eje imaginario, es decir cuando las raíces de la ecuación característica (1+G(s)H(s) = 0), no estén en el semiplano derecho.

Entonces considerando el criterio de estabilidad de Nyquist, este relaciona la respuesta de frecuencia de lazo abierto G(jw)H(jw) a la cantidad de ceros y polos de 1+G(s)H(s) que hay en el semiplano derecho s.

Además hay que indicar que usualmente este análisis se lo realiza de manera gráfica.

Este método es útil debido a que al graficar las curvas de respuesta de lazo abierto, y aunque G(s)H(s) sean de mínima fase, se puede establecer la estabilidad absoluta o relativa del sistema en lazo cerrado, sin determinar los polos de lazo cerrado.

Es importante indicar que el criterio de estabilidad de Nyquist está basado en un teorema de la teoría de las variables complejas, (representación conforme).

### 1.3.1 Teorema de Representación Conforme de Cauchi

Sea una función F(s) compleja $F(s) = Re.F(s) + Im.F(s)$, en donde $s = \sigma + jw$, de tal manera que un contorno cerrado en el plano s se transforma en una curva cerrada en el plano F(s), lo cual puede ser demostrado.

A medida que un punto representativo recorre este contorno complejo en el plano s, en sentido horario, se producen un total de N rodeos en torno al origen en el plano F(s), siendo $N = Z - P$, en donde Z es el número de ceros de F(s) y P corresponde al número de polos de F(s), encerrados por el contorno.

Por lo tanto el número de circunvalaciones que da el contorno en el plano F(s) con respecto al origen es una función tanto del número de polos como de ceros existentes en el plano s dentro del contorno, pero si el

contorno en el plano s tiene igual número de ceros y polos, entonces N=0 y la curva cerrada en el plano F(s) no encierra al origen.

Además, para cada uno de los puntos en el plano s corresponde un solo punto en el plano F(s) excepto para los puntos singulares. Debe notarse además que la transformación del plano F(s) en el plano s, no necesariamente es unívoca, por lo tanto para un punto del plano F(s), puede corresponder más de un punto en el plano s.

Otras características importantes que se deben de tomar en cuenta para la transformación son las siguientes:

- El ángulo entre dos curvas en el plano F(s) es el mismo que tienen dichas curvas en el plano s, es decir los ángulos se mantienen, y también la dirección de las curvas en el punto de cruce.
- Hay que tomar en cuenta que los ceros de 1+G(s)H(s), corresponden también a los polos de lazo cerrado, denominados raíces, y que los polos de G(s)H(s) son los mismos de 1+G(s)H(s).

Como ejemplo de ese análisis, en la figura 1.32a esta esquematizada una curva en el plano s, y en la figura 1.32b aparece su correspondiente representación en el plano F(s).

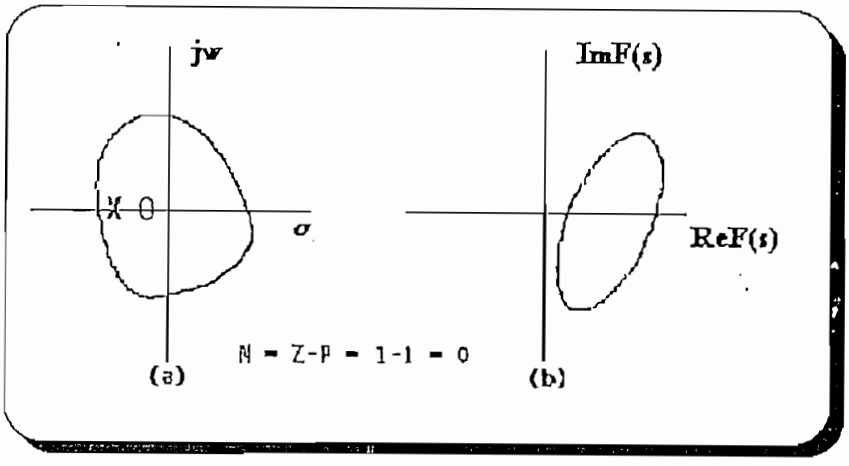



*Fig. 1.32 Ejemplo de representación conforme*

**1.3.2 Análisis de estabilidad usando la representación conforme**

Para realizar el análisis de estabilidad de sistemas de control lineal, se hace que el contorno cerrado del plano s abarque todo el semiplano derecho s; entonces el contorno va desde el eje jw=-∞ hasta cuando jw=+∞; cerrando el contorno con un paso circular de radio infinito en el semiplano derecho. Dicho contorno recibe el nombre de recorrido de Nyquist, el mismo que abarca entonces todos los ceros y polos de 1+G(s)H(s) que tengan parte real positiva.

Resulta necesario que el contorno de Nyquist no pase por ningún polo o cero de 1+G(s)H(s), para evitar esto se hace un pequeño rodeo por el polo o cero tal como se indicará más adelante.

Si se aplica el teorema de la representación al caso en que F(s) es igual a 1+G(s)H(s), entonces la cantidad de ceros en el semiplano derecho de F(s) es igual a la cantidad de polos de F(s) en el semiplano derecho de

s, más la cantidad de rodeos completos horarios al origen del plano 1+G(s)H(s).

El trazar G(jw)H(jw) para el recorrido de Nyquist es inmediato, más aún si se toma en cuenta que la representación del eje negativo de jw es la imagen simétrica del eje positivo jw respecto al eje real, es por ello que basta con graficar la curva para valores positivos de w, ya que para valores negativos será exactamente la misma rotada 180 grados respecto al eje real, y recorrida en sentido inverso.

Tal como se indicó anteriormente, para analizar estabilidad es necesario graficar la curva 1+G(jw)H(jw), y analizar las circunvalaciones alrededor del origen del plano complejo, pero se puede obtener exactamente el mismo resultado si se grafica G(jw)H(jw), pero en este caso se analiza el número de vueltas que de la curva en torno al punto -1+j0, tal como se aprecia en la figura 1.33

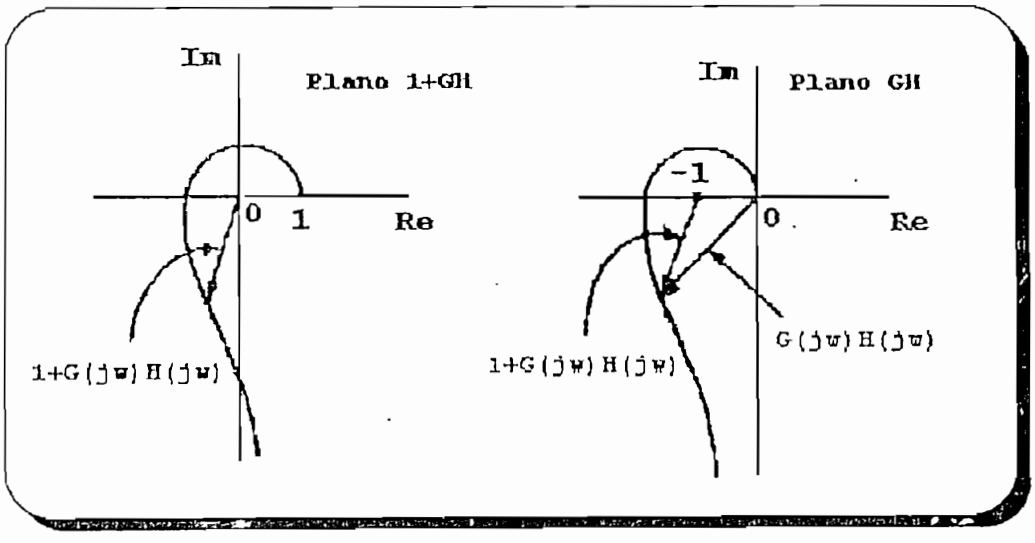



*Fig. 1.33 Diagramas de G(jw)H(jw) en los planos 1+GH y GH*

Considerando los rodeos al punto -1+j0, el criterio de estabilidad de Nyquist queda de la siguiente manera.

$$Z = N + P \qquad (1.28)$$

donde, como ya se indicó

Z = Cantidad de ceros de 1+G(s)H(s) en el semiplano derecho s

N = Cantidad de circunscripciones al punto -1+j0 en sentido horario

P = Cantidad de polos de G(s)H(s) en el semiplano derecho de s

Según el criterio de Nyquist para que haya estabilidad, no deben haber rodeos del punto -1+j0; Por lo tanto sólo con una pequeña porción para frecuencias positivas de la curva se puede establecer si existen o no rodeos al punto -1+j0, sin ser necesario dibujar la curva para todo el rango de frecuencia. Pero debe tenerse mucho cuidado cuando el sistema a graficarse sea de mínima fase, ya que para este tipo de sistemas es necesario graficar la curva tanto para frecuencias positivas como para negativas.

Como ya se indicó el recorrido de Nyquist no debe pasar por polos o ceros de G(s)H(s), por lo tanto si la función G(s)H(s) tiene polos o ceros en el origen o en el eje jw, se debe modificar el contorno en el plano s ya que de no hacerlo se hace indeterminada la representación de dichos puntos. Para realizar la modificación del contorno lo que se realiza normalmente es cambiar la vecindad del polo o cero que está en el eje jw, utilizando un semicírculo de radio infinitesimal $\varepsilon$ tal como se aprecia en la figura 1.34.

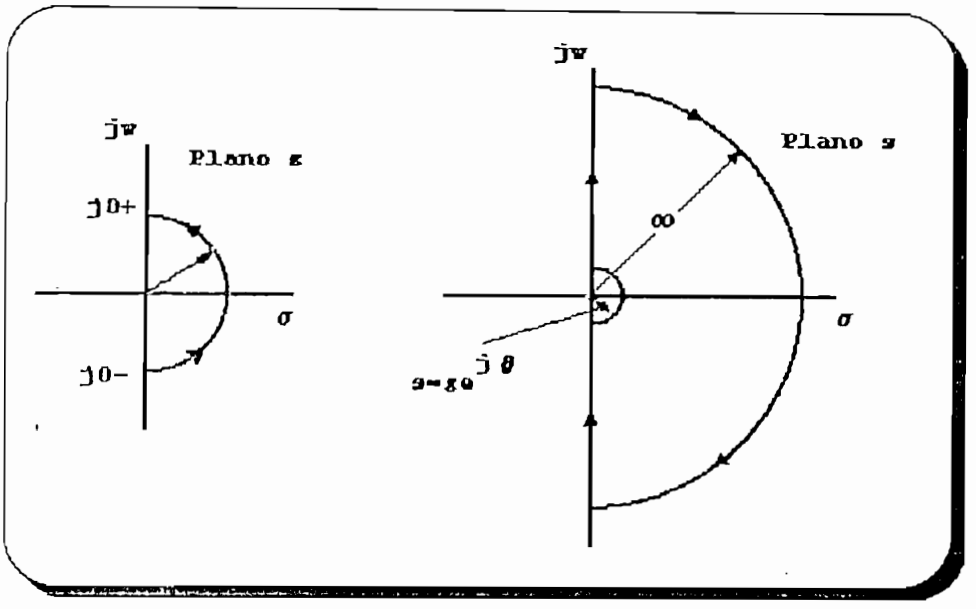



*Fig. 1.34 Contornos cerrados que evitan polos y ceros en el origen*

Como se aprecia en la figura anterior, se está eludiendo el centro del eje en donde podrían tenerse ya sean polos o ceros, lo cual se lo ha conseguido con un semicírculo de radio muy pequeño, entonces se garantizar que todos los polos y ceros que puedan existir en el semiplano derecho s, están contenidos en el contorno.

El desvío semicircular infinitesimal alrededor del origen se transforma en el plano GH en un círculo de radio infinito, lo cual puede demostrarse para una función de transferencia como la siguiente:

$$G(s)H(s)=\frac{K}{s(Ts+1)}$$

Como se puede apreciar, se tiene un polo en el centro del plano s, y la representación del semicírculo de radio infinitesimal sería:

$$s=\varepsilon e^{j\theta}$$

Donde θ varia desde -90° a +90°, por lo tanto G(s)H(s) en este punto resulta:

$$G(\varepsilon e^{j\theta})H(\varepsilon e^{j\theta})=\frac{k}{\varepsilon e^{j\theta}}=\frac{K}{\varepsilon}e^{-j\theta}$$

El valor de K/ε tiende a infinito al tender ε a cero, y $-\theta$ varía desde 90° a -90° cuando el punto representativo s recorre el semicírculo. Entonces los puntos $G(j0_-)H(j0_-)=j\infty$ y $G(j0_+)H(j0_+)=-j\infty$ están unidos por un semicírculo de radio infinito en el semiplano derecho de GH.

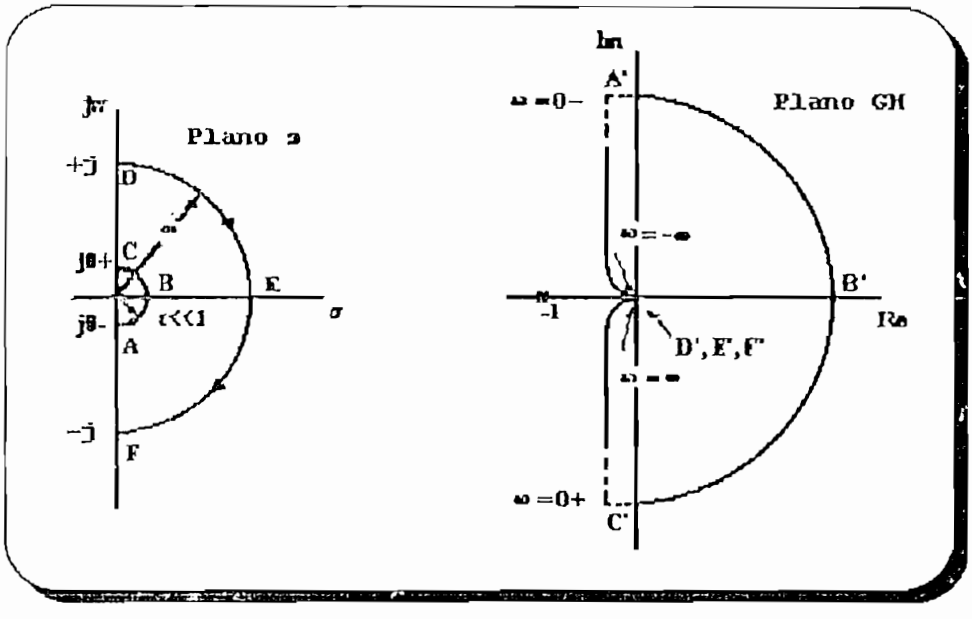



*Fig. 1.35 Contorno en el plano s y el lugar de G(s)H(s) en el plano GH donde G(s)H(s) = K/[s(Ts+1)]*

Lo anterior se observa en la figura 1.35; en donde los puntos A, B y C del contorno en el plano s, se representa por los puntos A', B' y C' en el lugar de G(s)H(s), mientras que los puntos D, E y F que están en el semicírculo de radio infinito en el plano s, se transforman en el origen del plano GH.

Se puede ver también en la figura 1.35 que no hay polos en el semiplano derecho s y el lugar de G(s)H(s) no encierra al punto -1+j0, entonces no hay ceros de la función 1+G(s)H(s) en el semiplano derecho de s, por lo tanto el sistema es estable.

Para el caso en que se tengan más de un polo en el origen, es decir cuando la función de transferencia de lazo abierto G(s)H(s) tenga un factor del tipo $1/s^n$; entonces el diagrama de G(s)H(s) tendrá n semicírculos de radio infinito, en sentido horario, alrededor del origen.

En cuanto al análisis de estabilidad que debe realizarse luego de tener el diagrama de Nyquist se debe anotar que valores negativos de N implican rodeos al punto -1+j0 en sentido antihorario.

Al examinar la estabilidad de los sistemas de control lineales usando el criterio de estabilidad de Nyquist, se ve que se dan tres posibilidades:

1.- No se encierra el punto -1+j0. Esto implica que el sistema es estable si no hay polos de G(s)H(s) en el semiplano derecho de s; en caso contrario, el sistema es inestable.

2.- Hay uno o varios rodeos antihorarios del punto -1+j0. En este caso el sistema es estable si la cantidad de rodeos antihorarios es igual a la de polos de G(s)H(s) en el semiplano derecho s; en caso de no serlo, el sistema es inestable.

3.- Hay uno o varios rodeos del punto -1+j0 en sentido horario. En este caso, el sistema es inestable.

### 1.3.3 Estabilidad Relativa Usando el Diagrama de Nyquist

Uno de los problemas importantes en el análisis de sistemas de control es el de hallar todos los polos de lazo cerrado o al menos aquellos más cercanos al eje jw que corresponden a los polos dominantes, si se conoce las características de respuesta de frecuencia de un sistema, resulta posible estimar estos polos de lazo cerrado más cercanos al eje jw; para ello se emplea una técnica totalmente gráfica basándose en la representación conforme del plano s al plano G(s).

Así por ejemplo si se considera la figura 1.36 en donde se ha dibujado curvas donde σ es constante mientras varía w, siendo s=σ+jw, y de igual manera, también están dibujadas curvas en donde es w quien permanece constante mientras cambia σ, y para cada una de estas familias de curvas se presenta su respectiva transformación en el plano G.

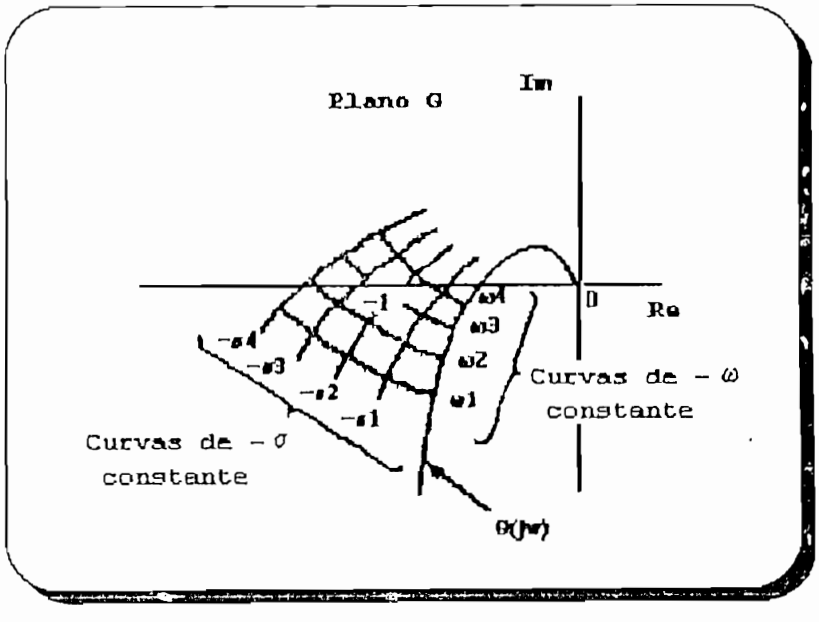



*Fig. 1.36 Transformación de una familia de curvas del plano s en el plano G(s)*

Se puede observar que en la figura 1.36 que la proximidad de G(jw) al punto -1+j0, constituye un parámetro indicativo de la estabilidad relativa de un sistema de mínima fase. En general, se espera que cuando más cercano se encuentre el lugar de G(jw) al punto -1+j0, se tendrá mayor sobreimpulso y tendencia a la inestabilidad. Por lo tanto cuando más cercanos están los polos de lazo cerrado al eje jw, más próximo estará el lugar de G(jw) al punto -1+j0

Si se dispone de un sistema que tenga al menos un par de polos de lazo cerrado complejos conjugados y, resulta que el punto -1+j0 está en la intersección de una curva de $\sigma$ constante con una de w igualmente constante en el plano G(s), estos valores determinados de $\sigma$ y w, definen respectivamente a $\sigma_c$ y $w_c$, y caracterizan el polo de lazo cerrado más cercano al eje jw.

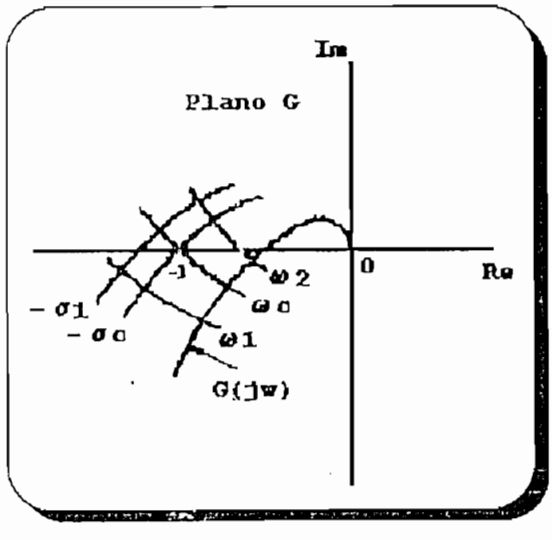



*Fig. 1.37 Estimación de $\sigma_c$ y $w_c$*

El diagrama de Nyquist a más de indicar la estabilidad o no de un sistema, da información del grado de estabilidad del mismo. Esto se consigue observando como ya se indicó la cercanía de G(jw) al punto -1+j0, pero dicha cercanía se la puede cuantificar con la ayuda tanto del Margen de Fase como de Ganancia.

**Margen de Fase.-** Es la cantidad de retardo de fase adicional necesaria a la frecuencia de cruce o de transición de ganancia para que el sistema quede al borde de la inestabilidad. La frecuencia de cruce de ganancia es aquella frecuencia para la cual el valor absoluto $|G(jw)|$ de la función de transferencia de lazo abierto es igual a 1. El margen de fase ($\gamma$) es 180° más el ángulo de fase ($\phi$) de la función de transferencia de lazo abierto a la frecuencia de cruce de ganancia, es decir $\gamma = 180 + \phi$.

En el diagrama de Nyquist se puede determinar rápidamente el margen de fase, trazando una línea desde el origen al punto en el cual un círculo de radio unidad cruza el lugar de G(jw), entonces el ángulo desde dicha línea al eje real negativo corresponde al margen de fase.

**Margen de Ganancia.-** Es la recíproca de $|G(jw)|$ a la frecuencia a la cual el ángulo de fase es -180°. Definiendo la frecuencia de cruce o de transición de fase $w_1$ como la frecuencia a la cual el ángulo de fase es -180°, da el margen de ganancia $K_g$:

$$K_g = \frac{1}{|G(jw_1)|}$$

Lo cual expresado en decibelios es igual a:

$$K_g db = 20 \log K_g = -20 \log |G(jw_1)|$$

El margen de fase expresado en decibelios es positivo si $K_g$ es mayor que la unidad y negativo si es menor que la unidad.

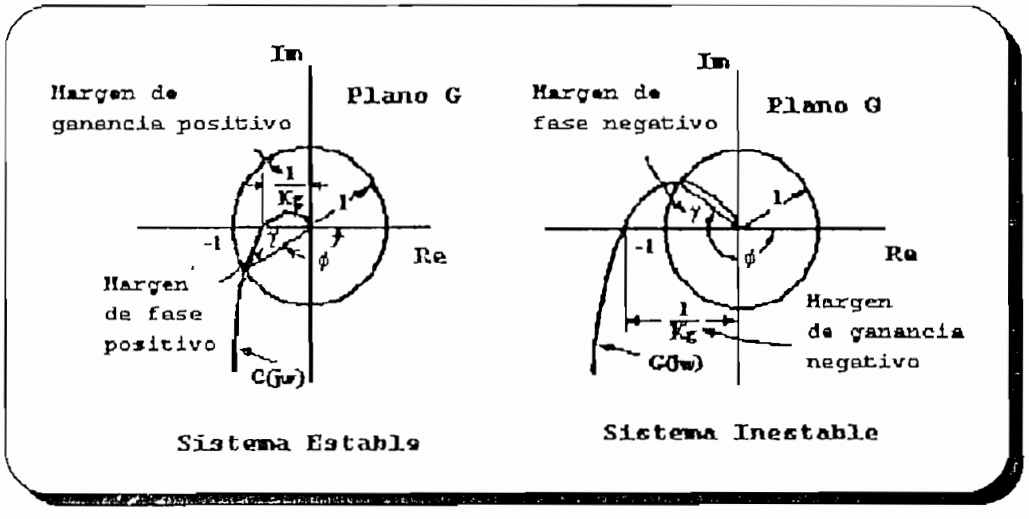



*Fig. 1.38(a) Diagramas Polares*

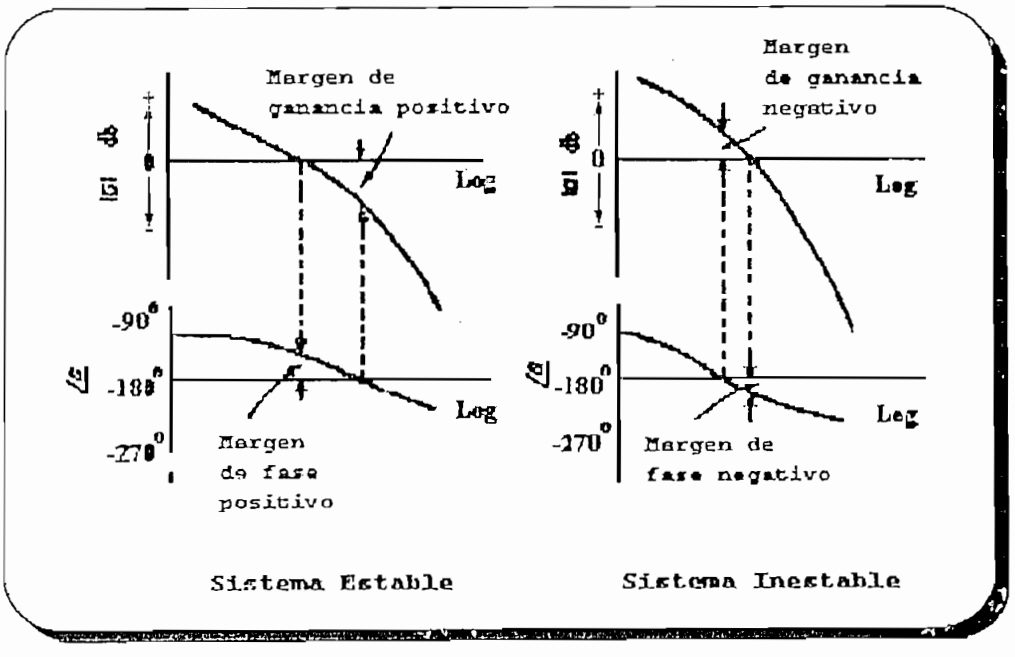



*Fig. 1.38(b) Diagrama Logarítmico*

*Fig. 1.38 Márgenes de fase y ganancia en sistemas estables e inestables*

La figura 1.38 muestra el margen de fase y de ganancia tanto para un sistema estable como para uno inestable

Es conveniente anotar que los márgenes de Fase y de Ganancia de un sistema de control son una medida de la cercanía del diagrama polar al punto -1+j0. Por lo tanto se pueden usar como criterio de proyecto.

## 1.4 ANALISIS Y DISEÑO USANDO LA FUNCION DESCRIPTIVA Y CRITERIO DE NYQUIST

Para analizar la estabilidad de sistemas no lineales se puede utilizar un método gráfico, en donde por un lado se grafica G(jw), es decir el diagrama de Nyquist, y por otro se grafica la función descriptiva, que como ya se analizó anteriormente puede depender tanto de la amplitud como de la frecuencia de la señal de entrada.

Para explicar esto se hará referencia a la figura 1.39 en donde aparece un lazo de control que consta de una parte lineal y otra no lineal, que representan todo un proceso.

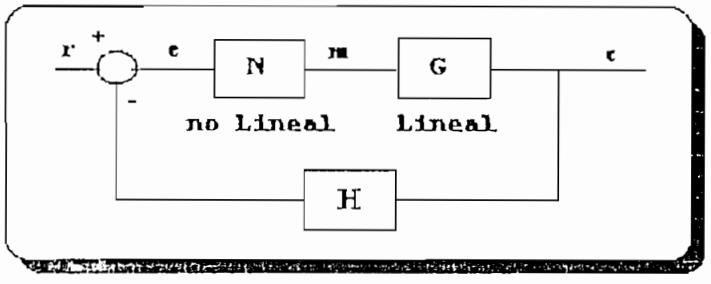



*Fig 1.39 Sistema de control no lineal*

El bloque N representa la función descriptiva de la parte no lineal del sistema, la misma que al atenuar suficientemente las armónicas superiores, se puede tratar como una ganancia de variable real o compleja. Entonces la función de lazo cerrado puede ser expresada como:

$$\frac{C(jw)}{R(jw)} = \frac{NG(jw)}{1+NG(jw)H(jw)}$$

La ecuación característica resulta entonces

$$1+NG(jw)H(jw)=0$$

$$G(jw)H(jw)=-1/N \qquad (1.29)$$

El desarrollo anterior como puede verse corresponde a un sistema que posee realimentación H(s), pero si la realimentación es unitaria, entonces la función de transferencia de lazo cerrado queda así:

$$\frac{C(jw)}{R(jw)}=\frac{NG(jw)}{1+NG(jw)}$$

Luego la función descriptiva para este caso será

$$1+NG(jw)=0$$

$$G(jw)=-1/N \qquad (1.30)$$

En todo el análisis posterior se considerará que la realimentación del sistema es unitaria, ya que de todas maneras siempre es posible reducir un sistema con realimentación no unitaria a uno con realimentación igual a uno, lo cual se puede ver en la figura 1.40

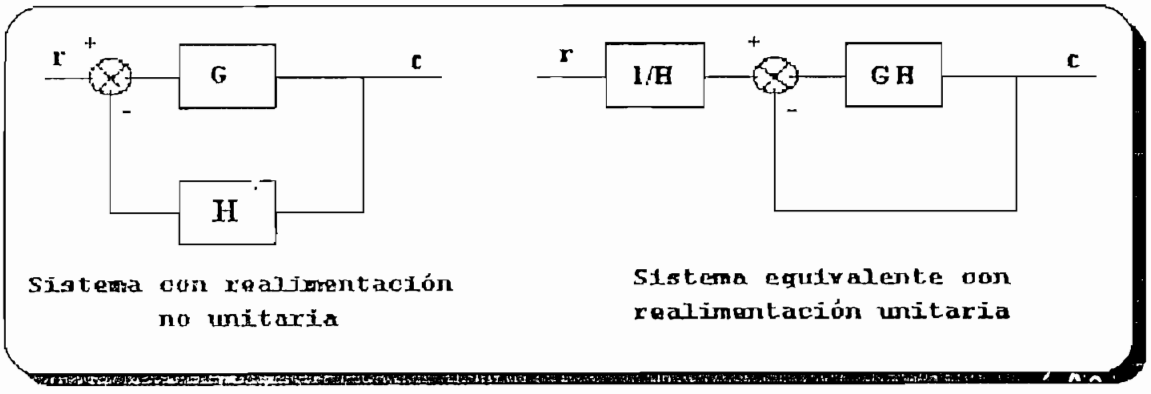



*Fig. 1.40 Transformación de un sistema con realimentación H(s) a uno con realimentación unitaria*

Como se puede notar, si se satisface la ecuación 1.29, equivale a que el lugar de G(jw) pase por un punto crítico, momento en el cual se encontraría al borde de la inestabilidad, por lo tanto se estaría hablando de un ciclo límite.

En el análisis de la función descriptiva se modifica el análisis convencional de respuesta de frecuencia de manera que todo el lugar de -1/N se convierte en un lugar de puntos críticos. En donde la ubicación del lugar de -1/N y G(jw) dan información respecto a la estabilidad.

Para establecer la estabilidad del sistema se trazan los lugares de -1/N y G(jw), y se considera que si el lugar de -1/N no se encuentra rodeado por el lugar de G(jw), el sistema es estable y por lo tanto no hay ciclo límite en estado estacionario.

Pero si el lugar de -1/N está rodeado por el lugar de G(jw), entonces el sistema es inestable.

También puede darse el caso de que exista intersección entre los lugares de -1/N y G(jw), entonces la salida puede presentar oscilaciones mantenidas o un ciclo límite

Pueden darse casos en los cuales se disponga de más de un punto de intersección entre los lugares de -1/N y G(jw), esto se puede apreciar en la figura 1.41

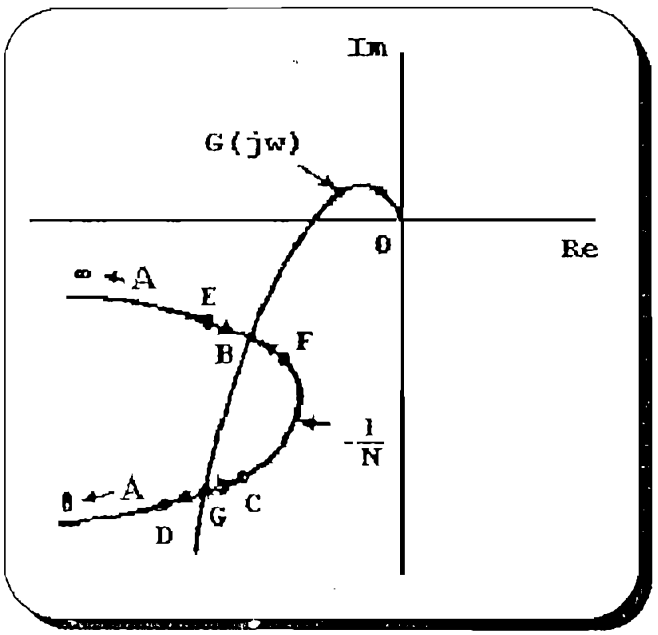



*Fig. 1.41 Análisis de estabilidad de ciclo límite*

En este caso hay que determinar cual de los dos puntos de intersección G o B corresponde al ciclo límite estable, para ello se hace lo siguiente.

Se debe analizar primeramente cual es el sentido de crecimiento de la amplitud (A) de la señal de entrada en la curva de -1/N, y como puede verse en la figura 1.40, el crecimiento se da desde el punto G hacia el punto B, por lo tanto, si originalmente el sistema está funcionando en el punto G, y por cualquier circunstancia se da una leve perturbación al sistema de modo que aumente la amplitud de la entrada al elemento alineal, lo cual ocasionaría que el punto G se desplace al punto C, y como puede observarse dicho punto está rodeado por la curva G(jw), por lo tanto como ya se indicó anteriormente se trata de un punto inestable, razón por la cual la amplitud tenderá a crecer más por encontrarse en un punto inestable y crecerá hasta llegar al punto B.

Por otro lado si la perturbación que se da al sistema que está trabajando en el punto G hace disminuir la amplitud de la entrada sinusoidal al elemento no lineal, situándose el punto de operación en el punto D, y como este punto no está rodeado por la curva G(jw), es un punto estable por lo tanto la amplitud tiende a disminuir alejándose del punto D hasta alcanzar la estabilidad total, por lo tanto el punto G posee características divergentes y corresponde a un ciclo límite inestable, es por ello también que se dice que el punto G presenta un equilibrio divergente.

A continuación se analiza lo que sucede al dar una leve perturbación al sistema estando trabajando en el punto B.

Por ejemplo se supone que se desplaza el punto de funcionamiento hasta el punto E, el mismo que al no estar rodeado por la curva G(jw) (punto estable), hace que disminuya la amplitud de la señal llegando nuevamente el punto representativo a localizarse en el punto B, por el contrario si la perturbación desplaza al punto B hasta el punto F, el mismo que al estar rodeado por la curva G(jw) constituye un punto inestable, que hace que la amplitud aumente llegando nuevamente hasta el punto B. Entonces el punto B posee características convergentes y el funcionamiento del sistema en el punto B es estable, en otras palabras, el ciclo límite en este punto es estable y se habla de un punto con equilibrio convergente.

Si se presenta el caso en el cual la curva de G(jw) no rodea a -1/N, se dice que el sistema es estable y no hay ciclo límite en estado

estacionario, por lo cual cualquier perturbación que se presente en el sistema tenderá a desaparecer.

Pero si la curva G(jw) encierra totalmente a -1/N el sistema será inestable y cualquier perturbación que se presente causaría un aumento de la señal de salida indefinidamente hasta que algún sistema de protección detenga dicho crecimiento, aunque también puede darse que en lugar de crecer la señal de salida, esta disminuya hasta llegar a desaparecer.

Cuando se presenta intersección entre las dos curvas aparece como ya se indicó ciclos límite, de los cuales se puede determinar la amplitud y la frecuencia de oscilación igualando las dos ecuaciones G(jw) y -1/N en el punto de cruce, aunque dicho punto de cruce puede encontrárselo analíticamente, es más simple determinarlo en forma gráfica.

Si al graficar las dos curvas se encuentra que existen ciclos límite, se puede solucionar esto mediante compensación, con lo cual se cambia la forma de la gráfica de G(jw) pudiendo llegar a eludir los puntos de cruce de las dos curvas, pero se debe tener mucho cuidado con respecto al tipo de compensación que se puede usar ya que si se introduce en la planta un control en adelanto, se cambia la efectividad de la suposición que se realizó al comienzo de este análisis, de que las plantas son generalmente filtros pasa bajos, por lo tanto el poner un controlador en adelanto hace que la planta tienda a convertirse en un filtro pasa altos.

Otra cosa que se debe tomar en cuenta es que el método de la función descriptiva resulta más exacto cuando los sistemas son de mayor orden, por el mismo hecho de que constituyen mejores filtros que bloquean las armónicas que presentan las señales periódicas a la salida de las no-linealidades.

## 1.5 ESTRUCTURA DEL SISTEMA PROPUESTO

El sistema que se propone para poder analizar estabilidad de sistemas no lineales esta compuesto por una parte no lineal que lo constituye su función de descriptiva, seguida por una parte lineal representada por su respectiva función de transferencia, de la siguiente manera:

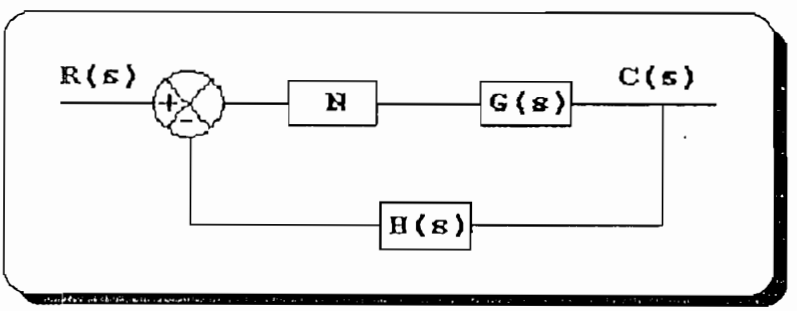



*Fig. 1.42 Sistema Propuesto*

Tal como se aprecia en la figura 1.42, G(s) representa la función de transferencia de la parte lineal del sistema que se va a analizar, mientras que N es la función descriptiva de la parte no lineal, pero también se ha colocado un bloque H(s) que representaría una realimentación no unitaria, pero como ya se indicó anteriormente todo sistema con realimentación no unitaria puede ser reducido a uno con realimentación unitaria.

El bloque G(s) representaría el equivalente que se puede obtener de la función de transferencia de la planta, ya que esta puede estar compuesta de un cierto número de bloques, con su respectiva función de transferencia, de igual manera N será el equivalente de las funciones descriptivas que

pueden presentar varias no linealidades colocadas una a continuación de otra.

Luego se puede encontrar una función equivalente de lazo cerrado de todo el sistema indicado en le figura 1.42 así:

$$F=\frac{G(s)*N}{1+G(s)*H(s)*N}$$

Si se analiza la función anterior, se pueden tener raíces de 1+G(s)*H(S)*N=0 con parte real igual a cero, es decir que estén colocados en el eje jw, que como ya se indicó anteriormente, cuando esta relación se cumple, el sistema es oscilatorio, encontrándose en el límite de la inestabilidad.

Es por ello que en este sistema presentado se puede encontrar que en el punto de corte al eje imaginario, se cumple que:

$$G(s)*H(s)=-1/N \qquad (1.31)$$

Si consideramos que N no existe entonces la ecuación característica queda representada como:

$$1+G(s)*H(s)=0$$

$$G(s)*H(s)=-1$$

De la ecuación anterior se puede observar que al graficar G(jw)*H(jw) y analizar las circunvalaciones que da la curva al punto -1+j0, se puede

analizar estabilidad, lo cual ya se indicó anteriormente al analizar el criterio de estabilidad de Nyquist.

Por lo tanto de lo anterior se desprende que para el sistema propuesto (ecuación 1.31), se puede aplicar el criterio de estabilidad de Nyquist en donde se graficará por un lado G(jw)*H(jw), y por otro se lo hará con -1/N, analizándose por lo tanto la intersección de las dos curvas que se van a presentar, teniendo que encontrar posteriormente sus respectivos puntos de cruce así como también el análisis correspondiente a la porción de curva de -1/N que estaría contenida por la curva G(jw).

Mediante este análisis de las curvas se podrá determinar la estabilidad del sistema, determinándose los puntos de cruce de las dos curvas, y analizando si se tratan de puntos estables o inestables, (método indicado anteriormente). Pero se debe tomar en cuenta que para un sistema no lineal el hecho de tener estabilidad en un punto determinado, no implica que el sistema no lineal sea estable, es por ello que en un sistema no lineal, estabilidad local no implica estabilidad global, entonces es necesario realizar el análisis de todos los puntos de la función descriptiva del sistema no lineal para determinar la estabilidad global.

Como se indicó anteriormente, en el sistema propuesto para el análisis, se puede involucrar compensadores, los cuales tienen como fin el ajustar la salida del sistema a un punto determinado, o de referencia. Para ello se compara el valor efectivo de la salida de la planta con el valor de la entrada, o punto de referencia para el sistema de control, se

determina entonces la diferencia entre estas dos magnitudes, y se produce una señal de control que trata de disminuir o reducir la desviación a cero.

Entre los tipos de compensaciones más conocidas se encuentran, la proporcional, proporcional derivativa, proporcional integral y proporcional integral derivativa.

Lo que sucede cuando se coloca un compensador en un sistema no lineal, es que la característica no lineal no es afectada por la compensación sino que igual como pasa en los sistema lineales, esta etapa de compensación solamente traslada el gráfico de la respuesta inicial del sistema.

Así por ejemplo al tener un compensador en serie con una no linealidad figura 1.43, se puede determinar su ecuación característica como:

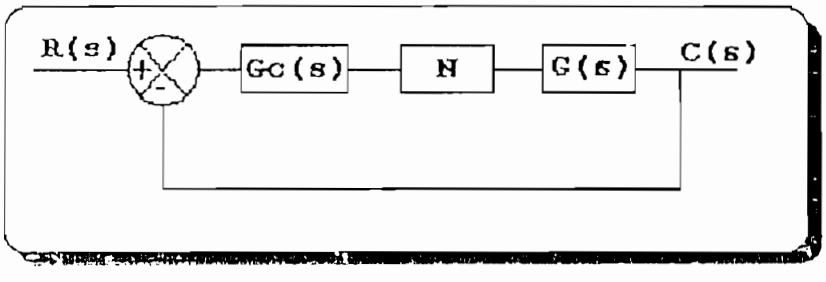



*Fig. 1.43 Sistema no lineal con un compensador.*

$$1+Gc*G*N = 0$$

$$Gc*G=-1/N \qquad (1.32)$$

Se puede observar de la ecuación anterior que al introducir la etapa de compensación equivale a recorrer el lugar de G(jw) a Gc(jw)*G(jw), pero el lugar de puntos de -1/N seguirá siendo el mismo.

Por lo tanto en el programa realizado para analizar estabilidad se deberá ingresar por separado, primeramente la función de transferencia de todo el sistema, es decir de la planta conjuntamente con los compensadores que tenga la misma, y por otro lado se ingresa el tipo de no linealidad a utilizarse, del mismo que se desprenderá su respectiva función descriptiva.

Debe tomarse muy en cuenta que los tipos de alinealidades empleados en este programa, son idealizaciones, ya que se está considerando que las alinealidades están formadas por rectas, lo cual es una aproximación, debido a que en la realidad son curvas, pero la aproximación es hecha lo más cerca posible a la realidad.

# CAPITULO II

## ANALISIS DE PROGRAMAS REALIZADOS

# CAPITULO II

## 2.1 ANALISIS PARA GRAFICAR -1/N, G(jw) Y LA SUPERPOSICION

Para la graficación tanto de la función descriptiva como de la de transferencia resulta necesario ingresar primeramente los datos de las respectivas funciones.

### 2.1.1 Ingreso de la Función de Transferencia

En la fig. 2.1 se indica como se ingresan los datos de la FT, en donde se aprecia que primeramente se coloca el grado del numerador y luego del denominador, realizándose enseguida una validación de los datos que se han ingresado, ya que el grado tanto del numerador como del denominador deben ser valores enteros, y entonces si el dato ingresado no corresponde a un valor entero, se escuchará una señal de alerta, e inmediatamente el programa volverá a pedir que se ingrese el dato.

Luego de haber ingresado el grado del denominador, el programa verifica que este valor sea menor o igual a 11, ya que este es el máximo grado permitido en el programa, pero si el grado del numerador resulta mayor que 11, nuevamente el programa pedirá que se ingrese tanto el grado del numerador como del denominador.

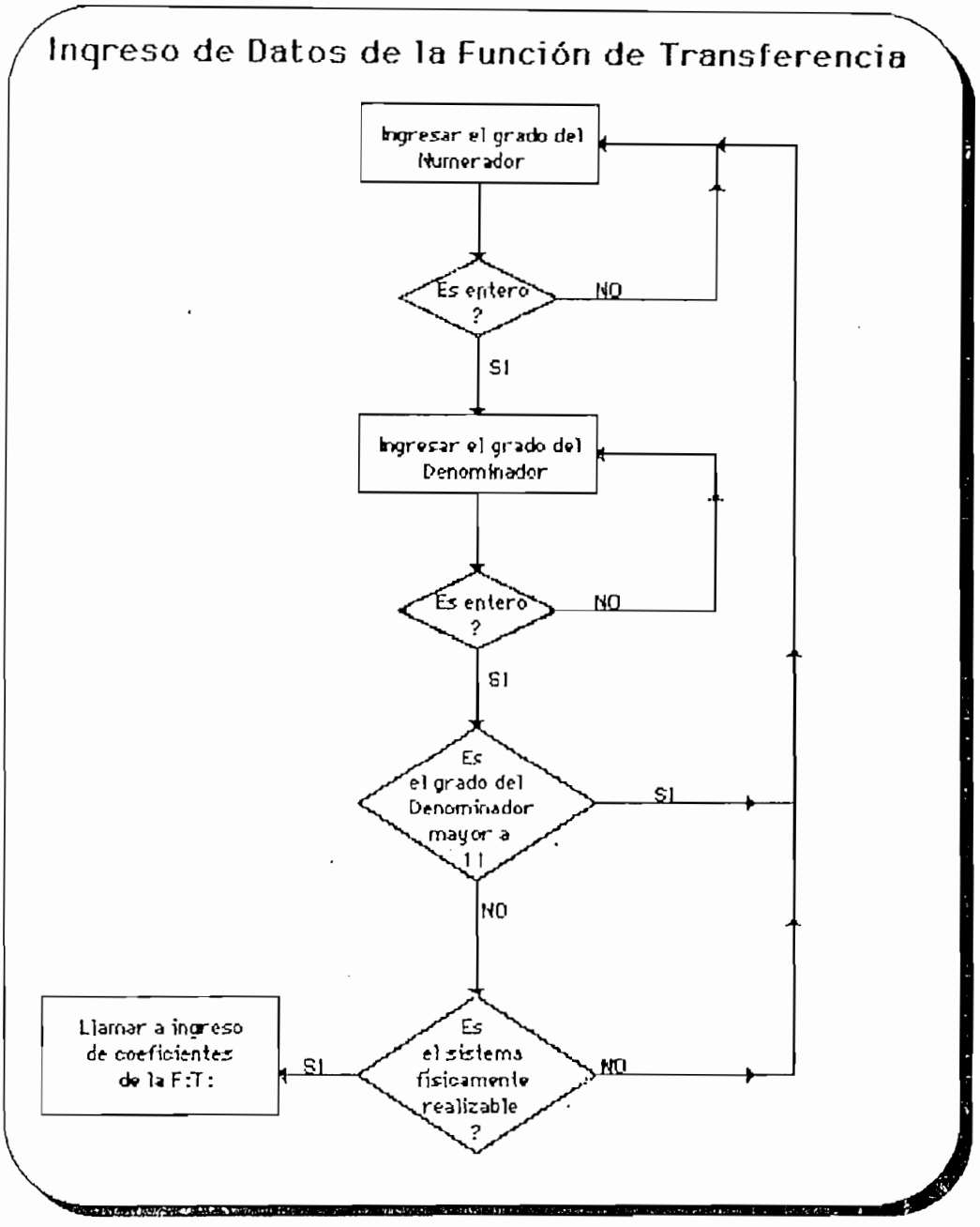



*Fig 2.1 Diagrama de flujo del ingreso del grado del numerador y del denominador*

El grado máximo del denominador que se ha puesto como límite se ha escogido buscando el mantener la estética de presentación de los coeficientes dentro de la pantalla, y aunque se podría poner un número mucho mayor para el grado del denominador, se saldría el sistema del sentido práctico que se le quiere dar, ya que sistemas de onceavo orden ya

es muy difícil encontrar en la realidad, y más aún un sistema de orden mayor; pero de encontrarse, se pueden aplicar los métodos de reducción de sistemas de órdenes elevados a uno de menor orden.

Continuando con el diagrama de flujo de la figura 2.1, luego de haber ingresado correctamente el grado del numerador como del denominador, se hace un análisis para ver si el sistema es físicamente realizable, es decir que el grado del denominador sea mayor al del numerador, y si esto no se cumple, se volverá a pedir que se ingrese el grado del numerador y del denominador.

Por último, si el sistema es físicamente realizable, se pasa a una subrutina aparte en donde se ingresan los coeficientes primero del numerador y luego del denominador de la función de transferencia, para lo cual hay que tomar en cuenta la forma general de la función de transferencia que es la siguiente.

$$FT=\frac{a_o+a_1*s+a_2*s^2+.....a_n*s^n}{b_o+b_1*s+b_2*s^2+.....bm*s^m} \qquad (2.1)$$

en donde:

n = grado del numerador

m = grado del denominador

La subrutina para el ingreso de los coeficientes de la función de transferencia se indica en la figura 2.2, en donde se puede apreciar, que existen dos diagramas de flujo totalmente independientes, pero muy similares, para el ingreso de los coeficientes del numerador y luego del denominador, diferenciándose únicamente las dos subrutinas, en que a los

coeficientes del numerador se los llama $a_i$, y a los del denominador $b_i$, que corresponde a la nomenclatura empleada en la ecuación 2.1.

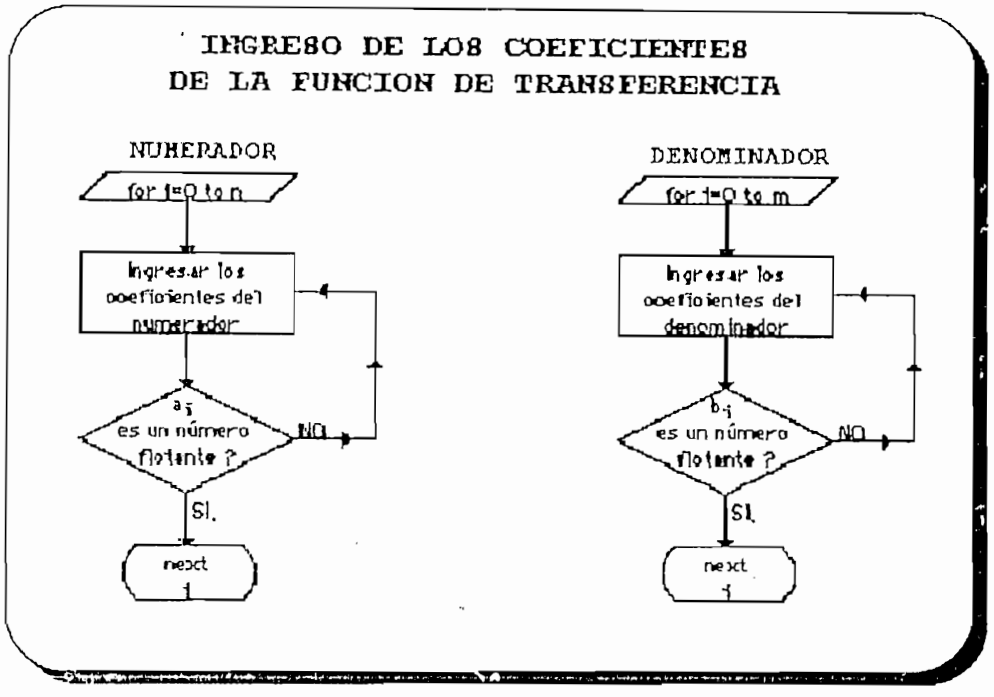



*Fig. 2.2 Diagrama de flujo del ingreso de los coeficientes de la FT.*

Analizando el diagrama de flujo se aprecia que también existe una validación de datos similar a la empleada en el ingreso del grado del numerador y del denominador, con la diferencia que en este caso se analiza si el dato ingresado corresponde a un número flotante o no, esto se lo hace con el fin de evitar que por error se coloquen por ejemplo letras en lugar de números, además se debe recordar que todo número entero es a su vez un número flotante. Además la validación que se realiza para los coeficientes del denominador es la misma que para el numerador.

Cuando se ha terminado de ingresar estos datos, se ha concluido con el ingreso de la función de transferencia, pero en este momento se debe analizar si los datos ingresados son o no consistentes para el algoritmo

empleado, ya que a continuación el programa obtiene una tabla de la parte real e imaginaria de la función de transferencia ante variaciones de la frecuencia w y con amplitud de la entrada constante, para ello el algoritmo empleado es el siguiente.

Se separa tanto el numerador como el denominador en una parte real y una imaginaria, para ello se reemplaza en la función de transferencia s por jw, y tomando en cuenta que $j^2=-1$, entonces se llega a una función equivalente de la siguiente forma.

$$FT=\frac{RealNum + jImagNum}{RealDen + jImagDen} = \frac{RealNum + jImagNum}{RealDen + jImagDen} \times \frac{RealDen - jImagDen}{RealDen - jImagDen}$$

$$FT = \frac{(RealNum \times RealDen + ImagNum \times ImagDen) + j(RalDen \times ImagNum - RealNum \times ImagDen)}{RealDen^2 + ImagDen^2}$$

Como puede verse de la ecuación anterior, si los datos ingresados hacen que el denominador sea igual a cero, entonces se tendrá una división para cero lo que convierte al sistema en una indeterminación, en este caso, aparecerá un mensaje indicando que los datos ingresados no son correctos, y por lo tanto se los deberá ingresar nuevamente.

Si los datos son correctos es decir no producen una indeterminación en el algoritmo antes mencionado, quedará grabado en memoria de la computadora los valores correspondientes de la parte real e imaginaria de la función de transferencia para distintos valores de w, este proceso es necesario realizarlo, ya que para el análisis de estabilidad se requiere graficar la parte real de la función de transferencia en función de la

imaginaria, para distintos valores de w, que corresponde al gráfico de Nyquist.

Lo anteriormente indicado puede apreciarse en el diagrama siguiente.

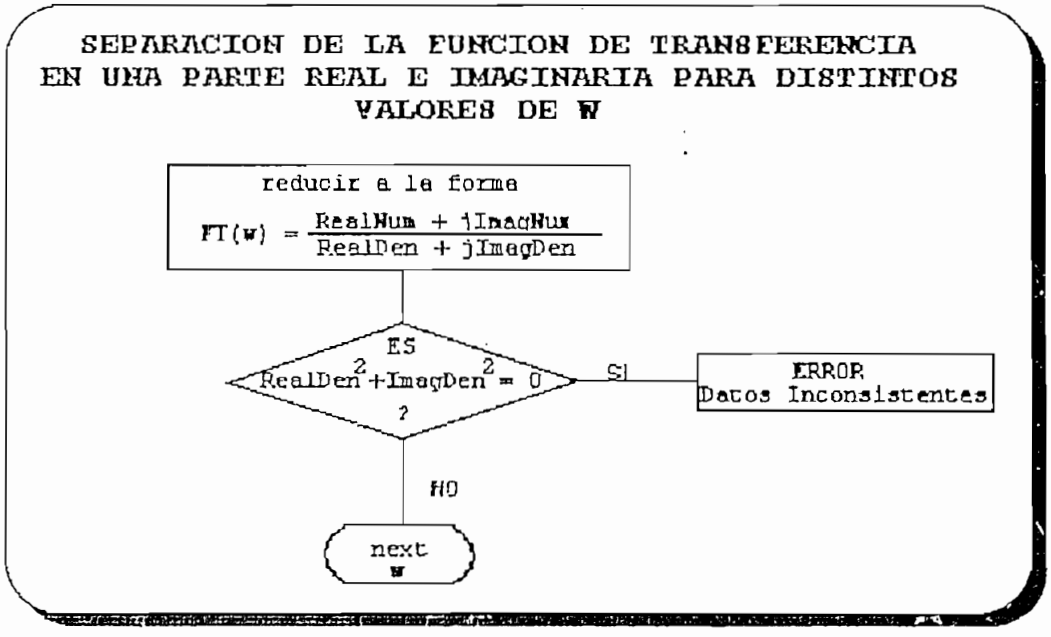



*Fig. 2.3 Diagrama de flujo de la validación de la FT. ingresada.*

Cabe destacar que estos datos obtenidos de la función de transferencia pueden también ser grabados en un archivo para luego poder presentarlos en la pantalla del computador, como también enviarlos a la impresora a través del pórtico Paralelo para algún otro uso; proceso que se indica en la parte concerniente a presentación de resultados.

Aparte de poder ingresar los datos manualmente, que se acaba de indicar, también existe la posibilidad de ingresar datos desde un archivo de resultados grabado anteriormente. Para ello, dicho archivo posee toda

la información concerniente a la función de transferencia que estaba empleándose en el momento en que se grabó el archivo.

Para poder emplear estos datos previamente grabados, solamente es necesario escoger en el menú principal de ingreso de la función de transferencia, que se van a usar datos ya grabados, e inmediatamente el programa pide el nombre de dicho archivo, y si no lo encuentra o no lo puede leer por alguna circunstancia, aparecerá un mensaje de error, pero si el archivo es leído correctamente, se procede a determinar la respuesta de frecuencia de la función de transferencia grabada, siendo exactamente el mismo proceso empleado cuando se ingresan datos nuevos, como se aprecia en la figura 2.3, por lo tanto si los datos contenidos en el archivo producen una indeterminación se presentará un error; en caso contrario, quedaría ingresada la función de transferencia grabada previamente en un archivo.

Para poder ver en la pantalla cual es la función de transferencia ingresada, existe una opción que permite revisar si la función ingresada es o no la que se desea analizar.

Luego de concluido todo este proceso se da por terminado el ingreso de la función de transferencia.

A continuación se va a indicar como se realiza el ingreso de datos de la función descriptiva, anotando que en el programa el ingreso de la una función (de transferencia) es totalmente independiente del ingreso de datos de la otra (función descriptiva), por lo tanto se podrá ingresar cualquiera

de ellas primero y luego la otra, aunque también es posible ingresar por ejemplo una función de transferencia determinada, luego una función descriptiva, y si se desea se puede cambiar ya sea por otra función descriptiva como por otra de transferencia las veces que sea necesario.

### 2.1.2 Ingreso de la Función Descriptiva

Para ingresar la función descriptiva se ha planteado el utilizar 12 alinealidades típicas, las mismas que se enumeran a continuación:

- Relés
  - Ideal
  - Con Zona Muerta
  - Con Pendiente
  - con Zona Muerta y Pendiente
- Saturación
  - Ideal
  - Con Zona Muerta
  - Con Pendiente
  - con Zona Muerta y Pendiente
- Histéresis
  - Ideal
  - Con Zona Muerta
  - Con Pendiente
  - con Zona Muerta y Pendiente

Como puede apreciarse, las doce alinealidades están dispuestas en tres grupos, por lo tanto se ha creado un menú que presenta a cada uno de ellos, de manera que primeramente se escoja uno de los tres grupos, y dentro del escogido aparece un submenú que permite escoger si la característica de la no linealidad deseada es Ideal, Con Zona Muerta, Con Pendiente, o Con Zona Muerta y Pendiente, en relación al tipo de no linealidad escogido previamente como es Relé, Saturación e Histéresis.

Esta forma de escogitamiento del tipo de no linealidad se indica en la figura siguiente.

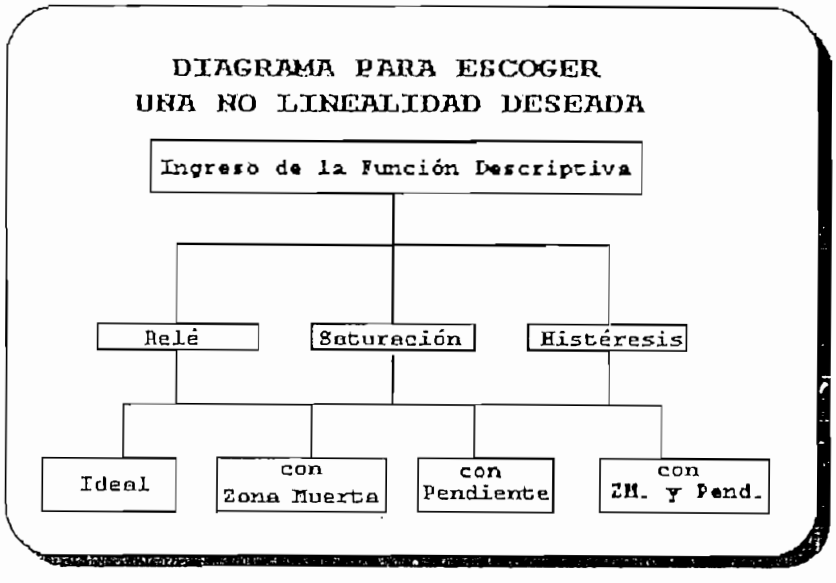



*Fig. 2.4 Diagrama de como escoger una no linealidad*

Tal como se aprecia en la figura anterior no se trata de un diagrama de flujo, solamente es un esquema de como escoger una no linealidad.

Luego de escogida la no linealidad que se desea analizar se pedirá el ingreso de sus respectivos coeficientes, correspondiendo una subrutina para cada uno de los tipos de no-linealidades, pero el algoritmo mismo de ingreso de coeficiente es el mismo para cada caso, lo que cambia es que para un tipo de alinealidad el número de coeficientes necesarios es distinto que en otro caso, para ello se presenta a continuación cada caso detalladamente.

*Relé Ideal*

Si se analiza la figura 1.4 se puede ver que sólo se necesita ingresar el valor de U para tener definido un relé ideal, con la única restricción de que U sea mayor que cero, ya que no tiene sentido el analizar el caso en que U=0 o U<0. Entonces el diagrama de flujo para este caso es el siguiente.

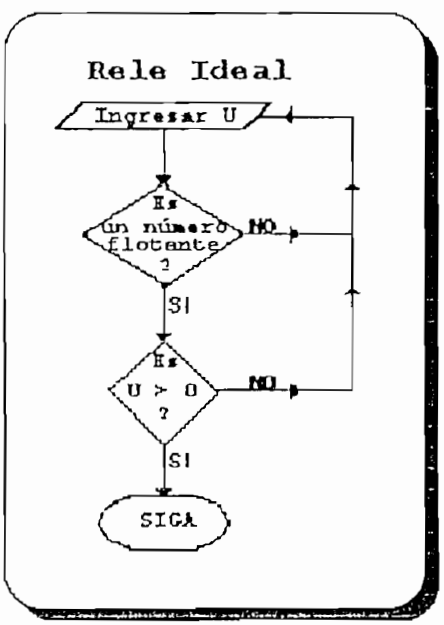



*Fig. 2.5 Diagrama de flujo para el ingreso de coeficientes de un relé ideal*

Como puede verse además de revisar si U es mayor que cero, también se hace una validación de datos para que no se ingresen valores raros como letras en lugar de números, este tipo de validación se hará con cada uno de los datos que se ingresen para la alinealidad que se escoja.

Al final del diagrama de flujo de la figura 2.5 hay un comando de siga, lo cual indica que en el programa se regresa al menú principal, pero en memoria ya están los datos de la función descriptiva escogida, este proceso es útil cuando luego de ingresada una no linealidad, se desea cambiar por otra.

*Relé con Zona Muerta*

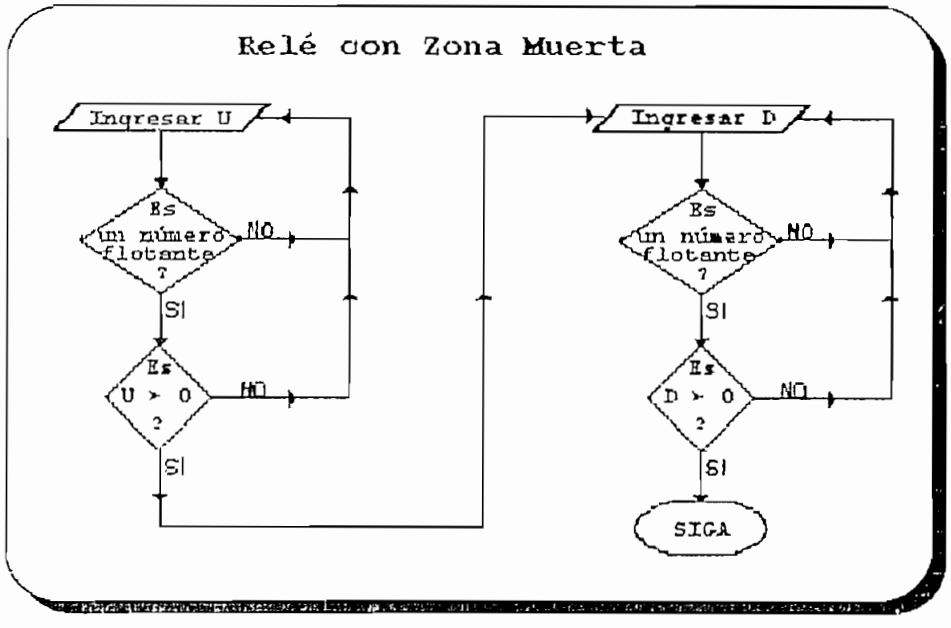



*Fig. 2.6 Ingreso de los coeficientes de un relé con Zona Muerta*

Para el ingreso de los datos de este tipo de no linealidad se emplea el diagrama de flujo de la figura 2.6, en donde como se aprecia se ingresa

valores de U y de D que son los datos que se requieren para definir totalmente a este tipo de relé, tal como puede verse en la figura 1.7.

Se tiene que la figura 2.6 es similar a la figura 2.5, ya que como se indicó la forma de ingreso de coeficientes de todos estos tipos de no-linealidades usadas en el programa son similares, luego de ingresar el valor de U se ingresa el valor de D, que debe ser también distinto de cero, ya que al ser D = 0, estaríamos nuevamente en el caso de un relé ideal.

*Relé con Pendiente*

Analizando la forma de un relé con pendiente, indicado en la figura 1.10, se observa que se requiere ingresar el valor de U como de K, para definir correctamente a este tipo de relé, siendo el correspondiente diagrama de flujo para este caso el indicado a continuación.

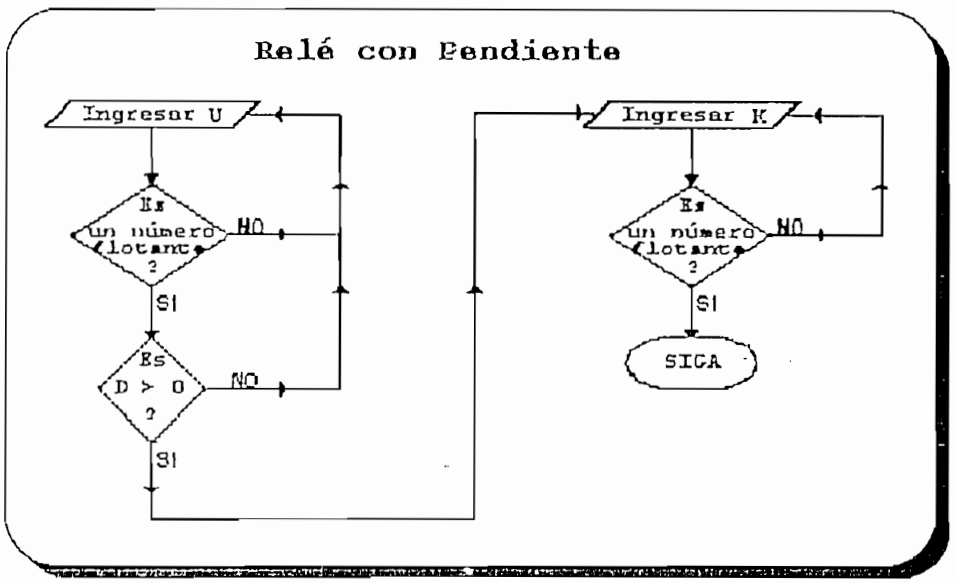



*Fig. 2.7 Ingreso de los coeficientes de un relé con Pendiente*

*Relé con Zona Muerta y Pendiente*

Este tipo de relé queda definido totalmente con el ingreso de los valores de U, K y D representados en la figura 1.12, e indicada su forma de ingreso en el diagrama de flujo de la figura 2.8

Se puede ver fácilmente que D no debe ser igual a cero ya que si esto ocurre, estamos regresando al caso de un relé con pendiente.

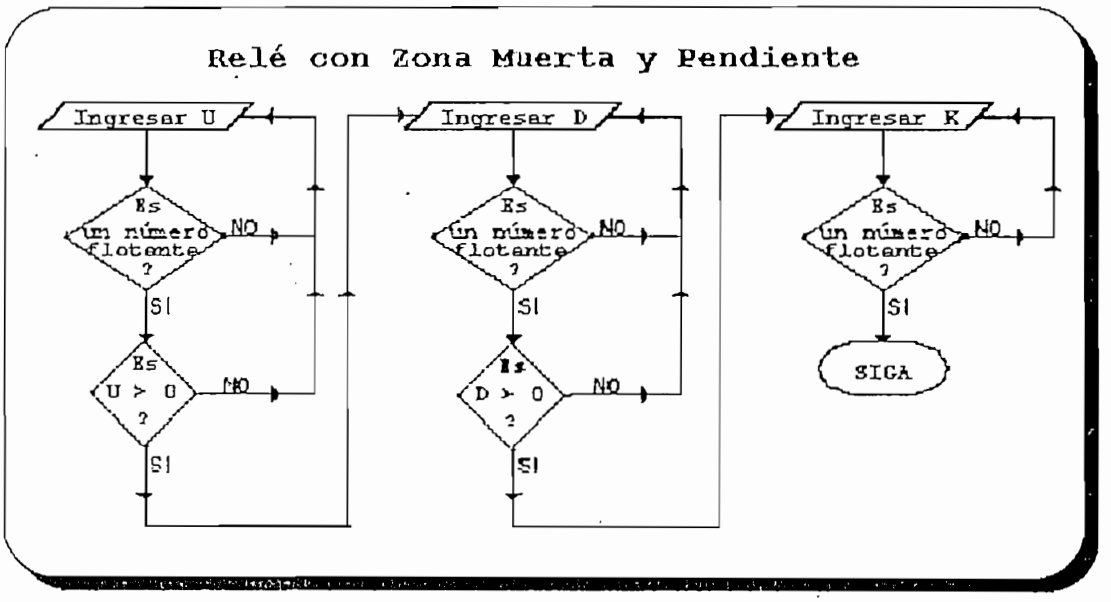



*Fig. 2.8 Ingreso de los coeficientes de un relé con ZM y Pend.*

Hasta ahora se ha analizado la forma de ingreso de los coeficientes de todos los tipos de relés mencionados en este capítulo, de aquí en adelante se indicará los diagramas de flujo para el ingreso de los coeficientes para los cuatro tipos de saturación como de histéresis indicados anteriormente.

*Saturación Ideal*

La forma de una saturación ideal está indicada en la figura 1.14, y su correspondiente diagrama de flujo para el ingreso de sus coeficientes es mismo de la figura 2.6 que corresponde a un relé con Zona Muerta, esto es debido a que para el relé con zona muerta es necesario ingresar el valor de U como de D para definir totalmente a este tipo de relé, pero exactamente los mismos coeficientes son lo necesarios para definir una saturación ideal, es por ello que el diagrama de flujo para el ingreso de datos de la saturación ideal es el mismo que para un relé con zona muerta, y lo que cambia es la ecuación que representa a la no linealidad, lo cual ya fue analizado en el capítulo anterior.

*Saturación con Zona Muerta*

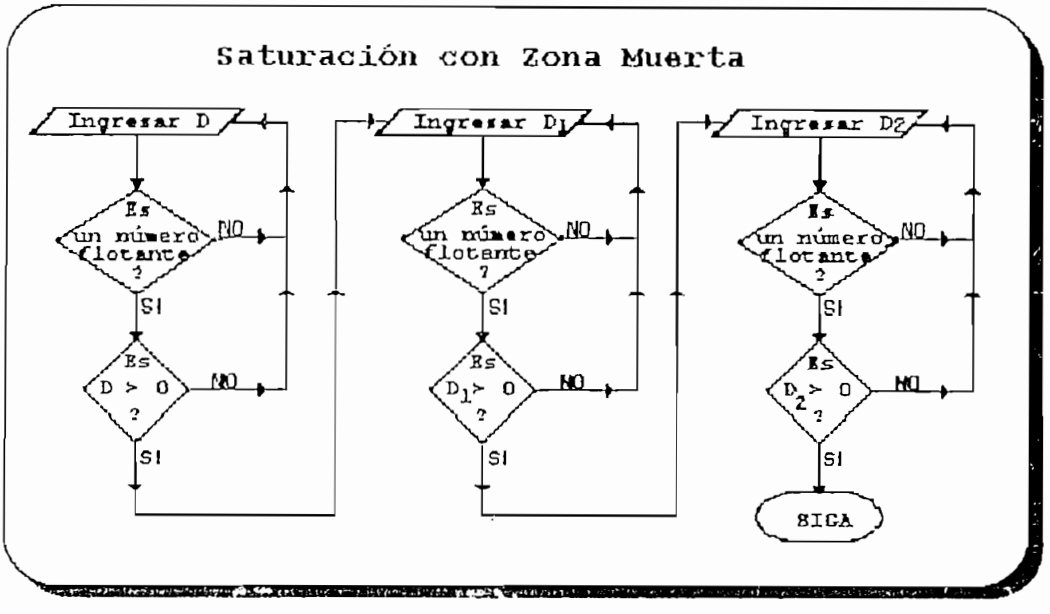



*Fig. 2.9 Ingreso de los coeficientes de una Saturación con Zona Muerta*

Una saturación con zona muerta, queda definida con los valores de U, $D_1$ y $D_2$, lo cual se puede ver en la figura 1.16, siendo entonces el

diagrama de la figura 2.9 el que determina la forma de ingreso de estos coeficientes.

Primeramente $D_1$ debe ser mayor que cero, mientras que $D_2$ debe ser mayor que $D_1$, lo cual es necesario debido a que si $D_1 = D_2$, estamos en el caso de un relé con zona muerta, y además $D_2$ no puede ser menor que $D_1$ debido a que produciría una indeterminación en las ecuaciones que representan la función descriptiva de este tipo de alinealidad.

*Saturación con Pendiente*

En este caso se requiere ingresar valores de U, D y K, para definir correctamente a este tipo de saturación como se indica en la figura 1.19, en donde $K_2$=K, mientras que $K_1$ queda determinado con U y D. pero como se puede ver los coeficientes que se requieren ingresar son los mismos que se ingresaron ya en el caso de un relé con zona muerta y pendiente, por lo tanto el diagrama de flujo del ingreso de dichos coeficientes será el mismo que se lo representa en la figura 2.8.

*Saturación con Zona Muerta y Pendiente*

Si se analiza la figura 1.21 que representa una saturación con zona muerta y pendiente se aprecia que se requieren los valores de U, $D_1$, $D_2$ y K, en donde $K_1$ queda determinado con $D_1$, $D_2$ y U, mientras que $K_2$ = K, entonces el diagrama de flujo para el ingreso de dichos coeficientes es el que se muestra en la figura 2.10

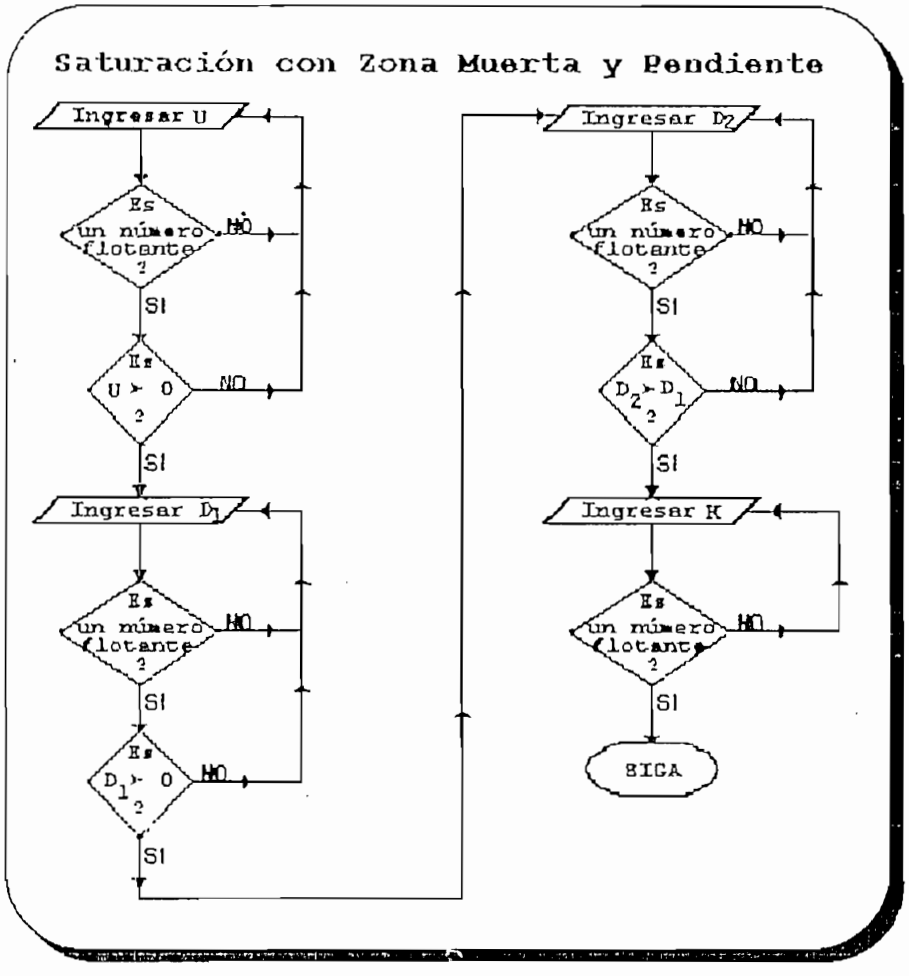



*fig. 2.10 Ingreso de los coeficientes de una saturación con ZM. y Pend.*

*Histéresis Ideal*

Para la histéresis ideal se requiere el ingreso de dos términos, que son el valor de U y D, figura 1.23, pero resulta igual que ingresar los datos ya sea de una saturación ideal como de un relé con zona muerta, los mismos que también quedan perfectamente definidos con los valores de U y de D, es por ello que el diagrama de flujo para el ingreso de los coeficientes de una histéresis ideal es el mismo de la figura 2.6.

*Histéresis con Zona Muerta*

Tal como se ve en la figura 1.25 para definir una histéresis con Zona Muerta es necesario ingresar los valores de U, $D_1$ y $D_2$, que son exactamente los mismos términos que se necesitan para definir una saturación con zona muerta, es por ello que el diagrama de flujo para este caso corresponde también al indicado en la figura 2.9.

*Histéresis con Pendiente*

Si se analiza la figura 1.27 se puede ver que también para este caso se requiere ingresar los valores de U, $D_1$ y $D_2$, es por ello que el diagrama de flujo para este caso también es el de la figura 2.9

*Histéresis con Zona Muerta y Pendiente*

Este caso ya es distinto a los anteriores debido a que se requiere ingresar 4 coeficientes que son, U, $D_1$, $D_2$ y $D_3$, lo cual se lo puede apreciar en la figura 1.29, según la cual debe cumplirse que $D_4>D_3>D_2>D_1$, y $U>0$, que corresponden a las únicas restricciones que debe presentar el diagrama de flujo para el ingreso de estos valores.

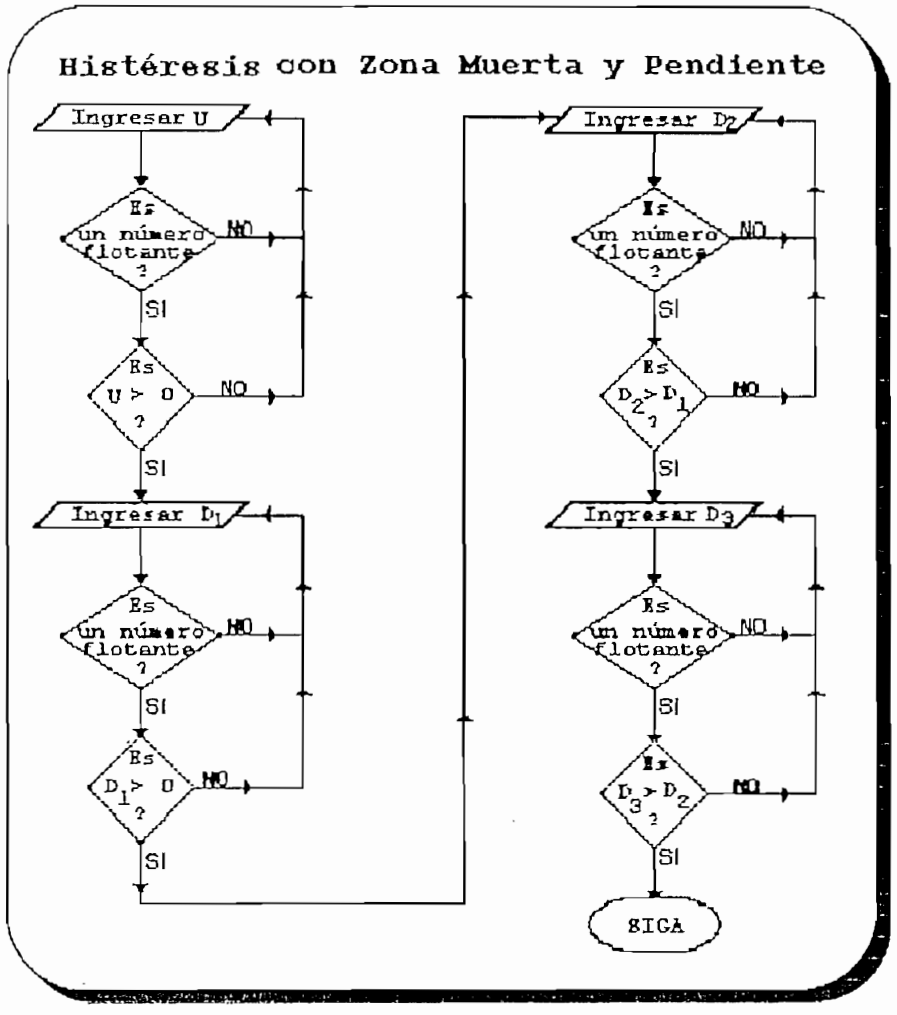



*Fig. 2.11 Ingreso de los coeficientes de una histéresis con ZM. y Pend.*

Con este análisis de la forma de ingreso de los coeficientes de las respectivas alinealidades, se da por terminado el ingreso de datos de las no-linealidades, entonces seguidamente de esto, el programa, hace una análisis de la función descriptiva ingresada, para lo cual se crea una tabla de valores en donde para variaciones de la amplitud de la señal de entrada (A), se determina la parte real e imaginaria de la función descriptiva, y dichos valores quedan en memoria del computador, pero también pueden ser enviados a un archivo, proceso que se indicará más adelante en la presentación de resultados.

Hasta este punto se ha llegado a indicar la forma de ingreso tanto de la función de transferencia como de la función descriptiva, pero se tiene la posibilidad de ingresar una compensación tipo PID, para lo cual el programa pide que se ingrese las constantes que determinan a un controlador PID, las mismas que son Kp, Kd y Ki.

El ingreso de las constantes del controlador PID se lo hace similar al ingreso de datos de la función de transferencia de la planta, esto es se verifica que los datos ingresados sean números flotantes, y que en ningún momento las tres constantes sean iguales a cero, ya que no tiene sentido el emplear una compensación nula.

Una vez ingresados los coeficientes de la compensación se hace un proceso para obtener la respuesta de frecuencia de G*Gc(jw), y como se pone la restricción de que para ingresar una compensación, primeramente se debió ingresar la función de transferencia de la planta, entonces cuando se ponen los datos de la compensación, ya se tiene calculada la respuesta de frecuencia de la planta grabada en dos vectores en la memoria del computador, los mismos que son realFT[jw] e imagFT[jw], con lo cual mediante el siguiente proceso se obtiene la respuesta de frecuencia de G*Gc(jw).

$$G(s) * G_c(s) = \frac{a_0+a_1s+a_2s^2+\ldots\ldots\ldots\ldots+a_ns^n}{b_0+b_1s+b_2s^2+\ldots\ldots\ldots\ldots+b_ms^m} \times Kp\left(1+\frac{Ki}{s}+Kd*s\right)$$

$$G(jw) * Gc(jw) = \left(realFT(w)+jimagFT(w)\right) \times \frac{Kp*(jw)+Kp*Ki+Kp*Kd(jw)^2}{jw}$$

$$G(jw) * Gc(jw) = \left[Kp \times realFT(w)+\left(\frac{Kp \times Ki}{w}-Kp \times Kd \times w\right)imagFT(w)\right] +$$

$$+ \quad j\left[\left(Kp \times Kd \times w - \frac{Kp \times Ki}{w}\right) realFT(w) + Kp \times imagFT(w)\right]$$

$$G(jw) * Gc(jw) = realFTc(w) + jimagFTc(w)$$

El resultado de la obtención de la respuesta de frecuencia para el sistema con compensación se lo graba en dos vectores que son realFTc(jw) e imagFTc(jw), en donde se puede apreciar que existe similitud con los vectores que guardan la respuesta de frecuencia de la planta sin compensación, una vez que se dispone de estos vectores se puede proceder a graficar la respuesta el diagrama de Nyquist ya sea de la planta sin compensación o con ella independientemente, es por ello que para obtener el diagrama de Nyquist de la planta sin compensación no es necesario ingresar una compensación, pero de haberla ingresado, también se puede obtener dicho gráfico.

Debido a la similitud de la respuesta de frecuencia del sistema sin compensación con el compensado, el proceso de graficación es el mismo, es decir se emplean las mismas subrutinas que se describen a continuación.

### *2.1.3 Graficación*

Dentro del proceso de graficación de la función de transferencia como de la función descriptiva, primeramente se realiza una verificación si se ha ingresado una función de transferencia y una alinealidad, o en el caso compensado de su respectiva compensación, ya que si no se ha ingresado alguna de estas, la parte correspondiente a graficación no se realiza, pero si se tienen ingresados almenos los dos

tipos de funciones, correspondientes a la planta y a la no-linealidad, entonces se procede a la graficación de un sistema de ejes cartesianos en todo lo ancho y alto de la pantalla del computador, luego se procede a graficar punto a punto la función de transferencia, la misma que ya se la ha almacenado en memoria, separada en su parte real e imaginaria, para diferentes valores de w, dichos valores están almacenados en dos vectores que son realFT[] e imagFT[], cuya longitud máxima es de 600, es decir se tendrán 600 puntos para la parte real de la FT, y sus correspondientes 600 puntos para la parte imaginaria, para el caso compensado estos vectores cambian de nombre a realFTc[] e imagFTc[], entonces como ya se indicó se hace una graficación punto a punto como se indica en la figura 2.12

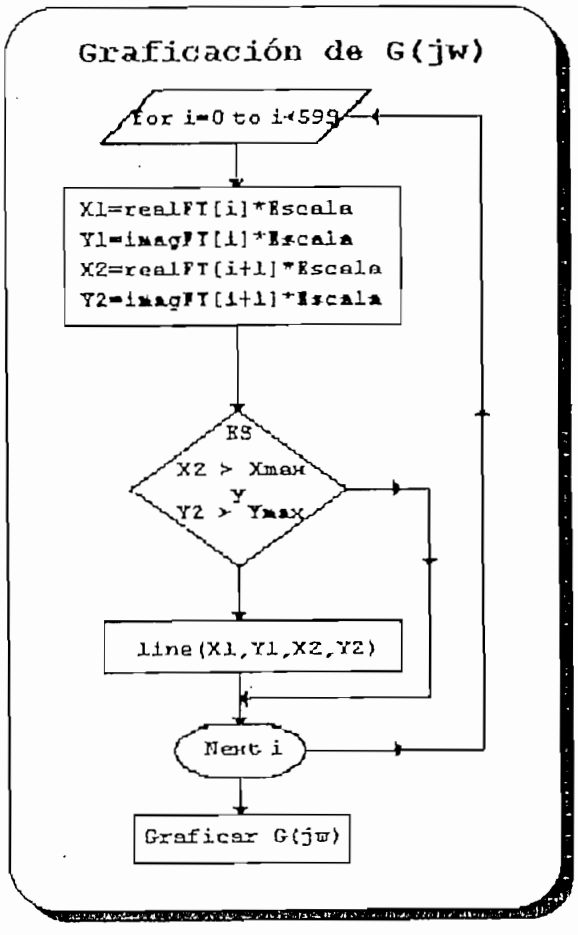



*Fig. 2.12 Diagrama de flujo para graficar la función de transferencia*

Pero se debe tomar en cuenta que se está haciendo una graficación a escala, la misma que puede ser cambiada, (proceso que se indicará más adelante), es por ello que se multiplica tanto la parte real de la función de Transferencia como la imaginaria por este valor denominado **escala**, pero además lo que se debe considerar que en la figura 2.12 se hace una comparación para ver si el punto a graficarse ya multiplicado por el factor de escala es mayor o no a un máximo que es el que se puede graficar en la pantalla del computador, de acuerdo al número máximo de pixeles de la misma, esto se lo hace para evitar que el programa siga graficando como si la pantalla fuera más grande, lo cual es un tiempo desperdiciado.

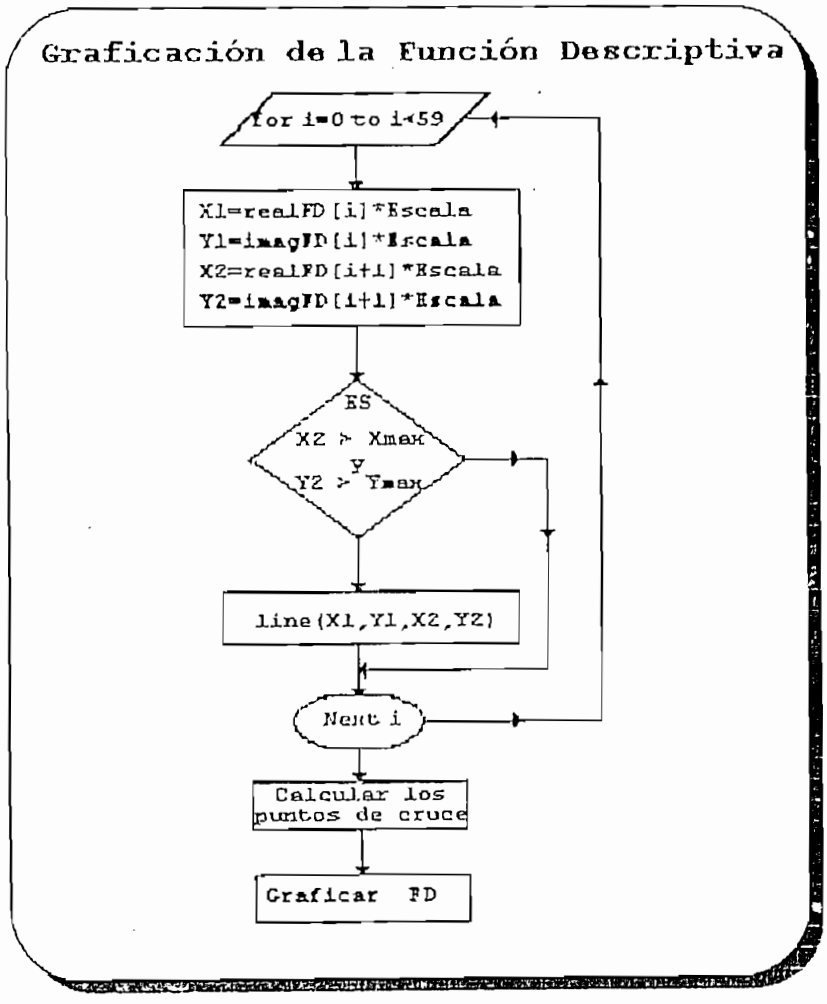



*Fig. 2.13 Diagrama de flujo para graficar la función descriptiva*

Con el diagrama de la figura 2.12 se indica como se grafica la función de transferencia, y una vez terminada dicha graficación se pasa a graficar la función descriptiva, que es un proceso similar, ya que en memoria se han ingresado dos vectores, uno con la parte real de la función descriptiva y otro con la imaginaria los cuales son realFD[] e imagFD[], en donde el número de puntos de estos vectores es de 60, entonces el correspondiente diagrama de flujo para este caso es el de la figura 2.13

Luego de graficada la función de transferencia y antes de graficar la descriptiva, se procede a encontrar los puntos de cruce de las dos curvas, lo cual es muy importante debido a que proporciona información de la estabilidad del sistema, para ello se emplea un algoritmo conocido como es el de Newton Raphson, mediante el cual se determina las coordenadas del punto de cruce, esto es la parte real e imaginaria, y luego con estos valores se reemplaza tanto en la ecuación que representa la función de transferencia como la función descriptiva, para encontrar los valores de la frecuencia (w), y la amplitud (A), en donde se produjeron dichas intersecciones de las curvas, esta determinación se lo hace mediante el empleo de otro algoritmo conocido para determinación de puntos de cruce como es el de regula falsi.

Entonces si se encuentra que existen puntos de cruce de las curvas, se calculan dichos puntos y se imprime en pantalla los valores de w y A en los cuales se produce la intersección, caso contrario se imprime un mensaje de que no existen puntos de cruce.

También se presenta un mensaje en pantalla de que se puede cambiar la escala mediante el siguiente procedimiento, si se presiona la tecla (+) se agranda la escala, pero si se presiona la tecla (-) se reduce la escala, este proceso aparece en la figura 2.14, en donde puede verse que si luego de graficada por primera vez, tanto la función de transferencia como la función descriptiva se presiona una tecla que no sea "+" o "-", el programa sale al menú principal, pero si se presiona una de estas teclas se cambia la escala, y se vuelve a llamar a las subrutinas para graficar tanto la función de transferencia como la función descriptiva, pero esta vez ya no se vuelve a calcular los puntos de cruce, ya que esto sólo se lo hace la primera vez, y en la veces siguientes cuando se cambia la escala, sólo se vuelven a presentar en la pantalla los valores de dichos puntos de cruce.

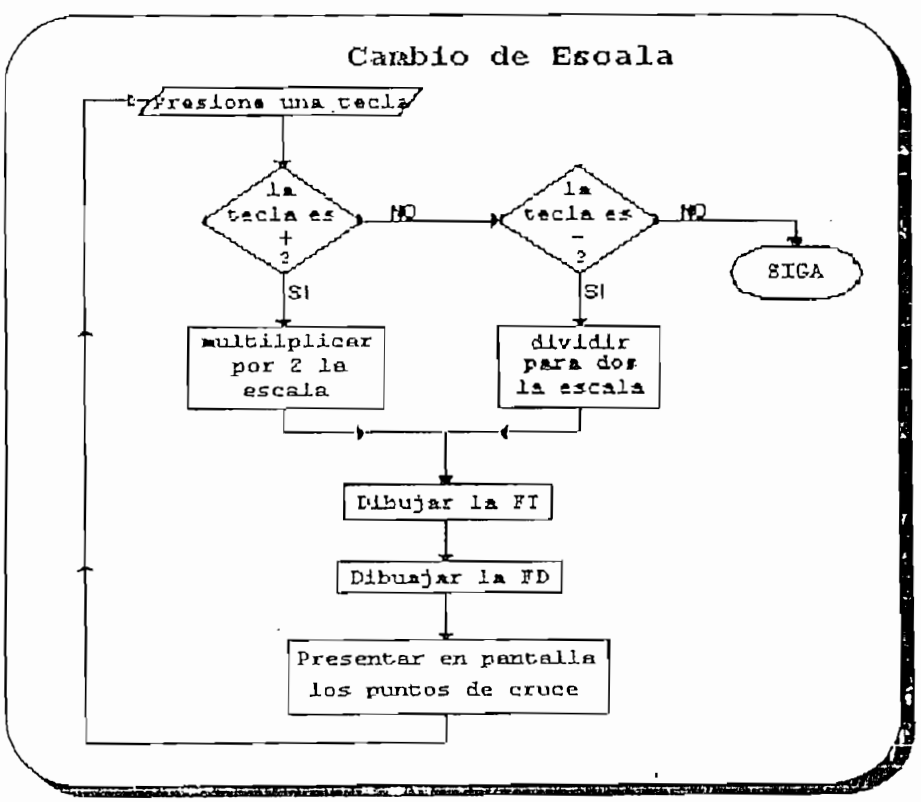



*Fig. 2.14 Diagrama de flujo para cambiar de escala*

## 2.2 PRESENTACION DE RESULTADOS

La presentación de resultados se realiza mediante una salida a la pantalla del computador, a un archivo, o a la impresora, los cuales son independientes entre si y serán analizados a continuación.

En cuanto a la presentación de resultados en la pantalla del computador, se tienen tres opciones, que son, el graficar las funciones -1/N y G(jW), lo cual sirve para análisis de estabilidad, también se puede graficar -1/N y Gc(jw), que corresponde al mismo sistema anterior al cual se le ha aumentado una compensación, además se puede graficar en la pantalla la respuesta del sistema en el tiempo; pero como complemento a la presentación de resultados se tiene opciones para ver en pantalla cual ha sido tanto la función de transferencia, como la no linealidad o la compensación ingresada.

Luego en caso de querer enviar los resultados a un archivo, se tiene cuatro posibilidades, independientes entre sí, pero relacionadas con la presentación de resultados en la pantalla del computador, esto es debido a que para poder grabar los resultados en un archivo, el programa exige que primeramente se haya graficado al menos una vez.

Los resultados que se pueden mandar a un archivo como se indicó son cuatro, la respuesta de frecuencia de la planta G(jw) o de la planta con compensación Gc(jw), como también de la no linealidad escogida (función Descriptiva) y de la respuesta en el tiempo del sistema.

Los mismos datos que se envían a un archivo pueden también ir a la impresora, obteniéndose los mismos resultados que aparecerían en el archivo tipo ASCII.

Como complemento a la presentación de resultados, se tiene que si uno ha grabado los resultados en un archivo, sin necesidad de salirse del programa se puede abrir dichos archivos y mostrarlos en pantalla, lo cual es una gran ayuda debido a que se puede analizar el comportamiento punto a punto del sistema y ver características importantes que talvez no se aprecian bien en el gráfico.

Pero por ser un archivo ASCII el que guarda la información de cualquiera de las respuestas de frecuencia o en el tiempo, los resultados pueden ser importados en una hoja electrónica como Lotus, Qpro o Excel, para futuros usos que se desee dar.

Como se puede deducir de la explicación anterior, el proceso de grabar resultados en un archivo, sea cual sea los resultados escogidos es similar, es por ello que se presenta en la figura 2.15 un diagrama de flujo de como guardar los resultados solamente de la respuesta de frecuencia G(jw). En dicha figura se aprecia que se pide ingresar el nombre del archivo pero sin extensión, esto es por que en el programa se encuentran reservadas las extensiones para cada tipo de dato que se guarde en un archivo, esto es para evitar la posibilidad de que se graben datos encima de otros, lo cual ocasionaría la pérdida de los mismos. Además se observa que se realiza una verificación para comprobar si el nombre del archivo

ingresado es compatible con el sistema operativo DOS, estos es por ejemplo si se pone un nombre con espacios intercalados, entonces no se podrá crear dicho archivos y será rechazado por el programa, es necesario indicar que dicha verificación se la realiza cada vez que se desee grabar un archivo, esto se usa aparte de verificar si el nombre es compatible con el sistema para ver si no hay problemas como por ejemplo el disco lleno o con protección lo que impide que se cree dicho archivos.

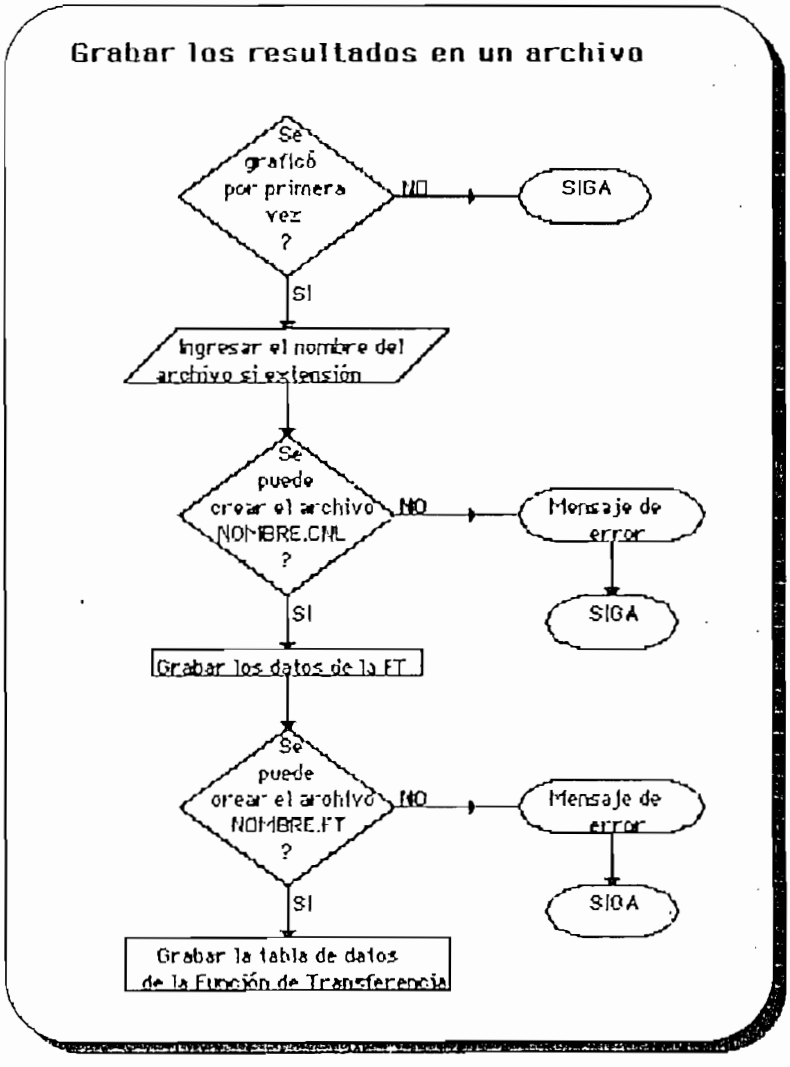



*Fig. 2.15 Diagrama de flujo para grabar los resultados en un archivo.*

Para el caso de grabar la respuesta de frecuencia de la planta sin compensación, como se aprecia en la figura 2.15 se crean dos archivos NOMBRE.CNL y NOMBRE.FT

Estos dos archivos son muy importantes, primeramente el archivo con la extensión FT es el que guarda la información de la función de transferencia de la planta, mientras que el con extensión CNL guarda datos importantes para el programa, lo cuales sirven cuando se emplea la opción de utilizar una función de transferencia grabada previamente.

Cosa similar ocurre cuando se graba la respuesta de frecuencia de la planta con compensación en donde se crean dos archivos, el uno con extensión FTC, que almacena la información de la respuesta de frecuencia de la planta con compensación, y el archivos con extensión GC que al igual que en el caso anterior guarda la información de la compensación utilizada, para poder recuperarla para otro uso posterior mediante la opción en el menú de utilizar una compensación grabada anteriormente.

En el caso de grabar los resultados de la función descriptiva o de la respuesta en el tiempo, solo se crea un archivo para cada uno de ellos, el mismo que tiene la extensión NOMBRE.FD para la función descriptiva y NOMBRE.TIM para la respuesta en el tiempo, en donde el primero guarda la información de la función descriptiva y el otro la respuesta en el tiempo.

Con lo indicado anteriormente se ha detallado la presentación de resultados en lo que tiene que ver con la grabación de estos en un archivo,

además como ya se mencionó se puede enviar los resultados a la impresora sin necesidad de grabar los resultados en un archivo, para ello se tiene el siguiente diagrama de flujo.

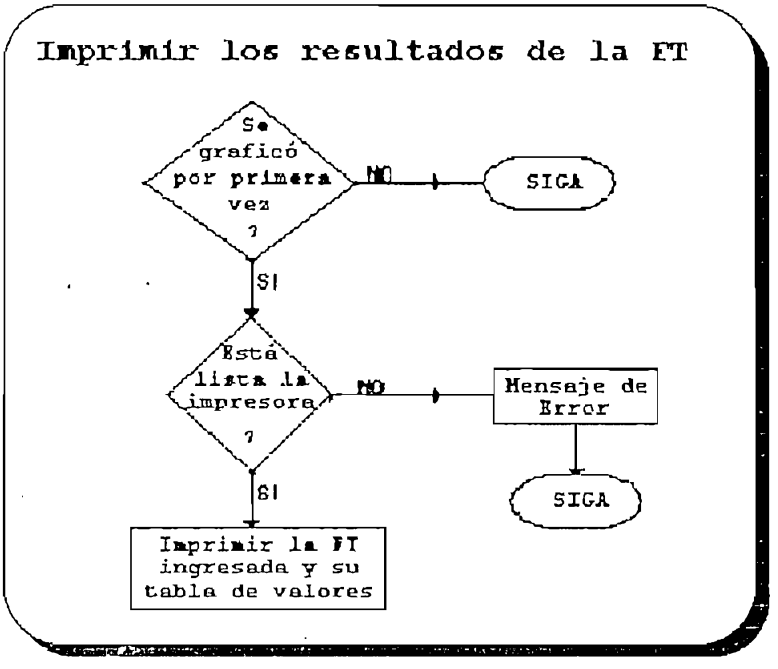



*Fig. 2.16 Diagrama de flujo para imprimir los resultados de la FT.*

Como se muestra en la figura 2.16 mientras que la impresora esté apagada o no tenga papel, o presenta algún problema detectado en la comunicación entre el computador y la impresora, no se podrá imprimir.

Además se debe indicar que en el momento de imprimir los resultados se tienen cuatro opciones que son los obtenidos mediante la función de transferencia de la planta, lo cual corresponde a su respuesta de frecuencia, proceso indicado en la figura 2.16, pero también se puede imprimir los resultado obtenidos con la compensación, si es que se ha escogido una, siendo el proceso el mismo de la figura 2.16. La tercera opción que se presenta de impresión de resultados corresponde a la función

descriptiva lo cual se indica en la figura 2.17. Y la cuarta opción que corresponde a la respuesta en el tiempo, que es un proceso similar a los anteriores.

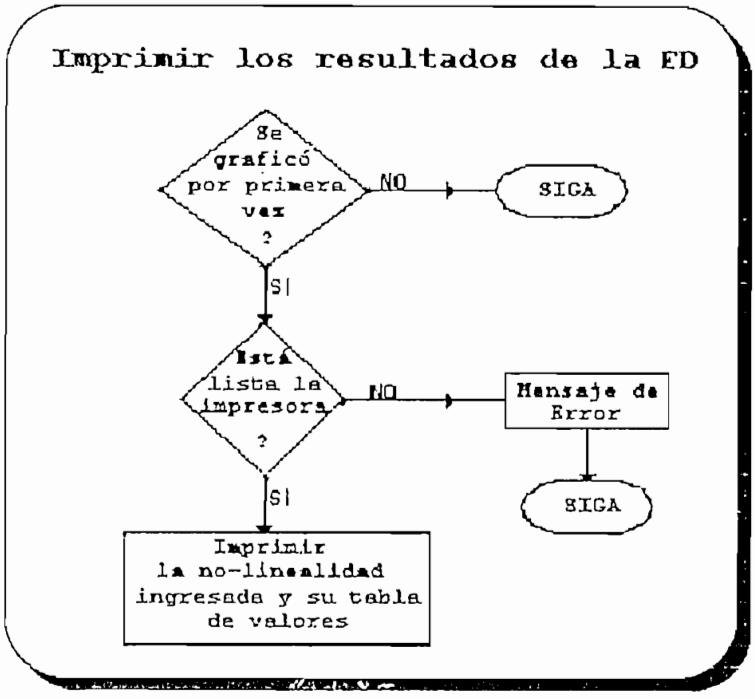



*Fig. 2.17 Diagrama para imprimir el tipo de no linealidad escogida, así como la tabla de valores de su FD.*

En las dos casos de impresión de resultados mostrados en la figura 2.16 y 2.17, se indica como se envía los resultados numéricos a la impresora, pero hasta ahora no se ha indicado como imprimir los gráficos que se obtienen, esto es por que la impresión de un gráfico es sumamente simple solamente con la opción de imprimir por pantalla que se encuentra en el teclado de la computadora.

Entonces con este análisis queda indicada la forma de presentación de resultados tanto en pantalla, como a un archivo y a la impresora, procesos que como ya se mencionó son independientes.

## 2.3 ESTABILIDAD

Tal como se mencionara anteriormente, la estabilidad de un sistema no lineal se la puede analizar por medio del diagrama de Nyquist de la función de transferencia de la planta, conjuntamente con el diagrama de -1/N que corresponde a la función descriptiva, pero para ello no basta simplemente con el dibujo de estas dos curvas, sino que se requiere para el análisis el encontrar los puntos de cruce de las mismas si es que los hay.

La forma de determinación de los puntos de cruce quedó indicada anteriormente, con la utilización del algoritmo de Newton Raphson, pero esto se lo hace tomando en cuenta que las curvas graficadas son la parte real en función de la imaginaria, pero lo que interesa es los valores de la frecuencia W como de la amplitud de la señal de entrada A para la cual se produce el punto de cruce de las dos curvas, para ello se hace un reemplazo de las coordenadas (x,y) del punto de cruce tanto en la función de transferencia como en la descriptiva para determinar la frecuencia W como la amplitud A respectivamente, siendo entonces necesario emplear otro algoritmo denominado de regula falsi. Este proceso de determinación de los valores de frecuencia y amplitud en el punto de cruce se lo muestra en el diagrama de flujo de la figura 2.18.

En dicha figura se aprecia que cuando se grafica por primera vez G(jw) y -1/N, se calcula el punto de cruce con los algoritmos indicados antes, pero si se cambia de escala, ya no es necesario volver a calcular los puntos de cruce, ya que estos no varían al cambiar de escala el

gráfico, en cuyo caso solamente se presentaran en pantalla los puntos de cruce ya calculados cuando se graficó por primera vez.

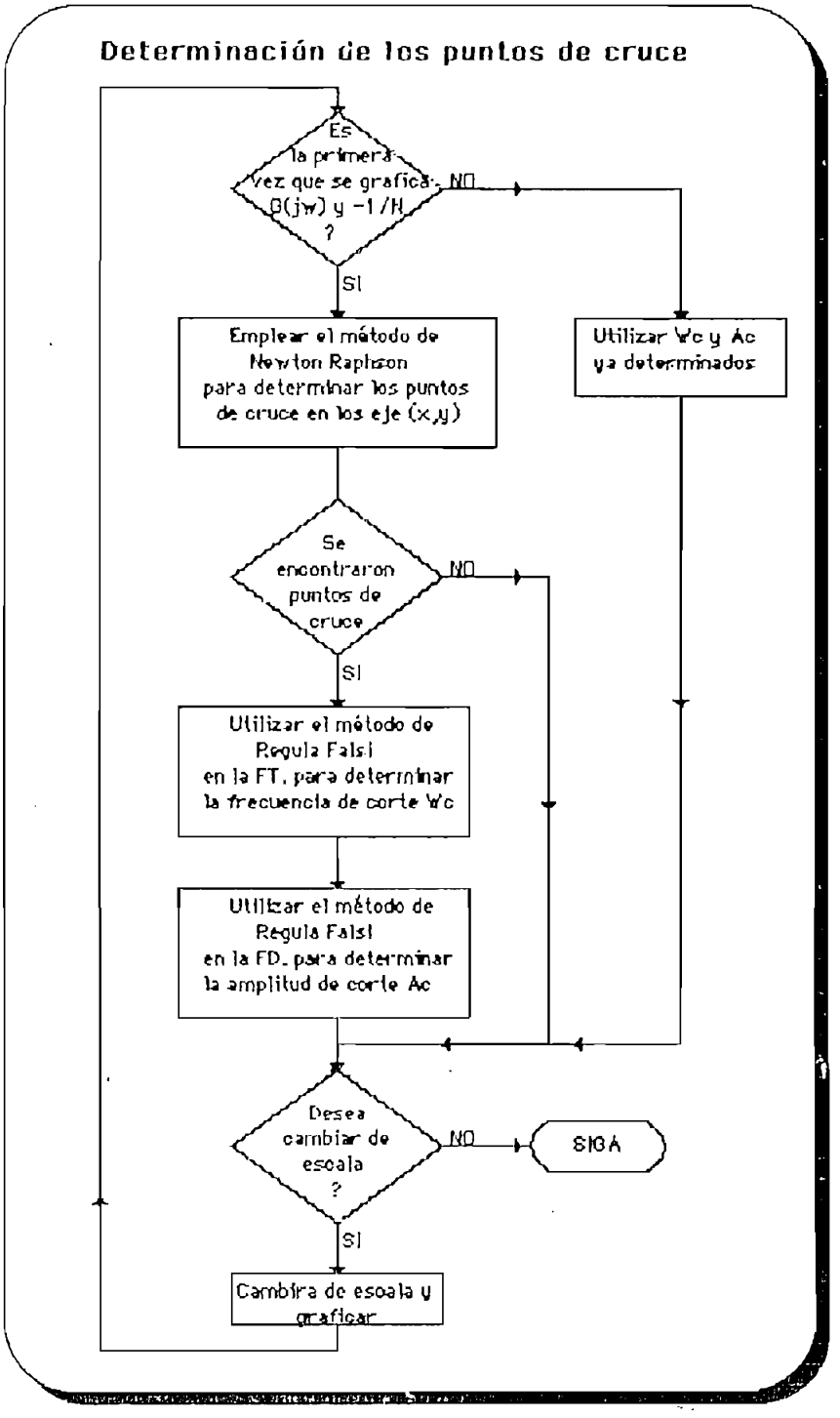



*Fig. 2.18 Diagrama de flujo de la determinación de los puntos de cruce*

Luego de obtener el gráfico en frecuencia, con la presentación en pantalla de los puntos de cruce, se puede ver si el o los puntos de cruce si es que existen, se tratan ya sea de un ciclo límite estable o inestable, esto es posible analizar viendo la forma como crece la amplitud de la señal de entrada A en el diagrama de -1/N, el mismo que se lo hace empezando con valores pequeños de A, y tendiendo hacia valores elevados, cosa similar ocurre con el gráfico de G(jw), ya que este se lo hace desde valores pequeños de w siguiendo en sentido ascendente el valor de la frecuencia w.

Entonces si uno puede darse cuenta de como va cada una de las curvas en función de sus variables, w y A respectivamente, se puede determinar visualmente si se trata de un punto de cruce estable o inestable, para ello es necesario recordar que toda porción de la curva -1/N que se encuentre dentro de la curva G(jw), corresponde a puntos inestables, según se analizó en el capítulo 1.4.

Pero si la graficación resulta muy rápida, cosa que dependerá del tipo de planta y de no linealidad empleada, puede que no se alcance a apreciar la forma como crece la curva en la pantalla, entonces se puede usar el siguiente método, primeramente grabar los resultados en un archivo, tanto de la respuesta de frecuencia de la planta sea esta con o sin compensación según se esté empleando, y también se debe grabar la respuesta de frecuencia de la no linealidad, que corresponde a su función descriptiva, y una vez grabados estos archivos, se lo puede abrir y presentarlos en pantalla, con lo cual se tendrá una tabla de valores de las dos curvas graficadas anteriormente, y como se dispone de los valores del

puntos de cruce, se puede hacer una análisis simple observando estos datos y determinar así si el punto de cruce es estable o inestable, según como tiendan los valores de W y A en las cercanías del puntos de cruce.

Con este análisis uno puede ver como se indicó si el sistema es o no estable, pero no se puede apreciar çuan estable es el sistema, es por ello que en el programa se tiene la posibilidad de obtener la respuesta en el tiempo del sistema, lo cual facilita para determinar que tan estable es el sistema, este proceso se lo indicará posteriormente en el numeral 2.5 donde se analiza el método de simulación empleado.

## 2.4 DISEÑO

Mediante este programa se puede diseñar como se va a emplear una compensación, ya que teniendo la planta, se puede escoger la no linealidad que uno quiera, o que disponga conjuntamente con la planta, y con ello se procedería a graficar la respuesta de frecuencia de G(jw) como de la función descriptiva de la no linealidad -1/N, proceso mediante el cual se puede determinar los puntos de cruce, y ver si el sistema es estable o inestable en ciertos valores, que dependerán de las características de las dos curvas, entonces, se puede proceder a buscar una compensación de tal manera que se disminuya o elimine la porción de -1/N que esté contenida en la curva G(jw), ya que dicha porción de curva representaría los puntos inestables del sistema, y según la compensación que se emplee, se puede graficar la respuesta de frecuencia de la planta compensada y comparar con la de la planta sin compensación, para determinar entonces si se ha logrado eliminar este problema de inestabilidad, en ello consiste el proceso de diseño, para lo cual la compensación que se puede emplear es del tipo PID, en donde uno ingresa de acuerdo a su criterio los valores de Kp, Ki, Kd.

La determinación de estas coeficientes del compensador PID, pueden ser obtenidos mediante varios métodos como son los de lazo abierto, de Ziegler Nichols o de los ajustes de Pemberton, pero para ello es necesario graficar la respuesta en el tiempo de la planta en lazo abierto, pero como el programa permite la graficación en el tiempo del sistema en lazo cerrado, se pueden emplear métodos de ajuste de lazo cerrado como el de Ziegler Nichols mostrado a continuación:

Considerando:

Ku = Ganancia que se le da al sistema para llevarlo a la oscilación

Tu = Período de la oscilación mostrada en el gráfico en el tiempo

Wc = 2*π/Tu

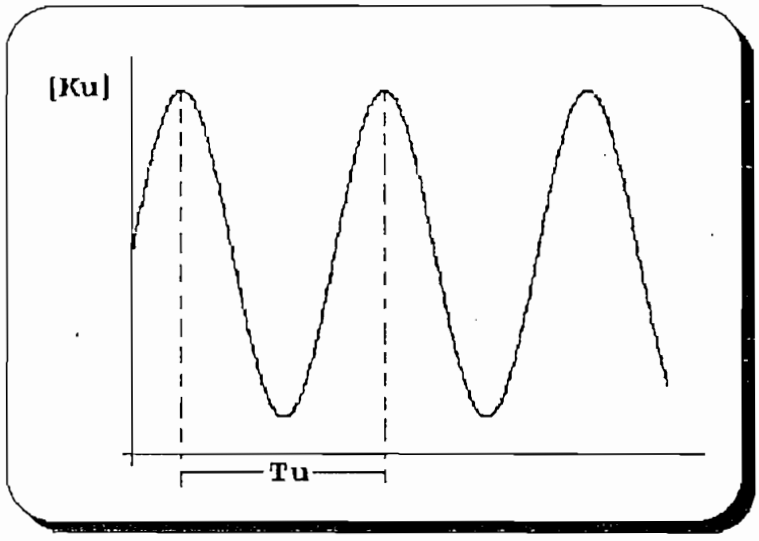



*Fig. 2.19 Sistema Oscilatorio*

| TIPO DE CONTROL | No Interactivo | Interactivo |
|---|---|---|
| P | Kp = 0.5*Ku | Kp = 0.5*Ku |
| PI | Kp = 0.45*Ku<br>Ti = Tu/1.2 | Kp = 0.45*Ku<br>Ti = Tu/1.2 |
| PID | Kp = Ku/1.7<br>Ti = Tu/2<br>Td = Tu/8 | Kp = Ku/3.4<br>Ti = Tu/4<br>Td = Tu/4 |

Pero debido a la posibilidad que presenta el programa para graficar el diagrama de Nyquist, se puede emplear el método de Åstrom que hace un

análisis en frecuencia, en donde se emplea los valores de Margen de Ganancia o de Fase para determinar las constantes del regulador PID.

$$Gc(S) = Kp\left[1 + \frac{1}{sTi} + sTd\right]$$

Entonces lo que nos interesa para aplicar este método es determinar cual sería el margen de Fase o de Ganancia que debe tener nuestro sistema, considerando solamente la planta con la compensación, independiente de la no linealidad, para que así la curva de G(jw) ya compensada no encierre a ningún punto de la curva -1/N que representa al tipo de no linealidad empleado.

Este método indica que los coeficientes del compensador se pueden obtener de la siguiente manera:

*a) CONOCIENDO EL MARGEN DE GANANCIA MG*

Definiendo a:

A = MG. especificado pero como magnitud (no en db)

$$Kp = Ku/A$$

$$Td = \frac{1}{wc^2 \times Ti}$$

En donde Ti se escoge arbitrariamente.

*b) CONOCIENDO EL MARGEN DE FASE MF*

si $\phi_{MF}$ = Margen de Fase

$$Kp = Ku*Cos(\phi_{MF})$$

$$Ti = 4*Td$$

$$Td = \frac{\tan(\phi_{MF}) + \sqrt{1+\tan^2(\phi_{MF})}}{2wc}$$

Pero como puede observarse en estos métodos, se requiere que el sistema sea oscilante en lazo cerrado, pero si esto no ocurre, se recomienda colocar un integrador más en la planta, o caso contrario se deben emplear los métodos de análisis en lazo abierto.

En lo que tiene que ver con la forma de ingreso de los coeficientes del PID en el programa es similar a la forma de ingreso de datos utilizada ya sea para la función de transferencia de la planta como de una linealidad.

También se debe destacar de que no se podrá ingresar un compensador si antes no se ha ingresado la función de transferencia de la planta, esto es por que una vez ingresado el compensador, el programa automáticamente calcula y mantiene en memoria del computador la respuesta de frecuencia del sistema compensado, es decir la planta más el compensador empleado.

## 2.5 SIMULACION

En el presente programa se tiene la posibilidad de realizar una simulación del comportamiento del sistema, para ello se emplea la graficación de la respuesta del mismo en el tiempo, graficación que puede realizarse tomando en cuenta varios factores.

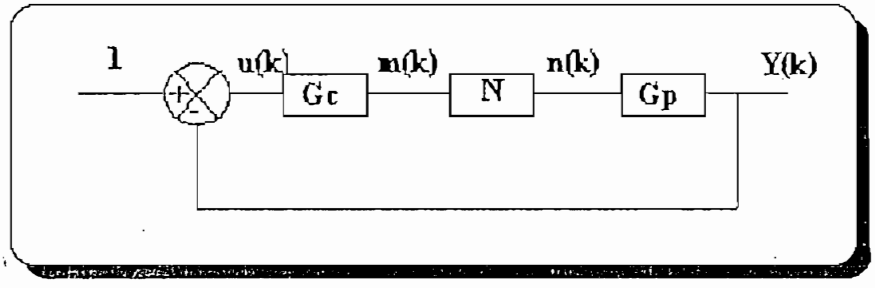



*Fig. 2.20 Sistema Propuesto para discretización*

Si se analiza la figura anterior, se puede ver que existen varias posibilidades de presentar el sistema a simular, como son por ejemplo, el tener solamente la planta sin ninguna compensación y sin alinealidad alguna lo cual facilita para obtener una respuesta en el tiempo de la planta, pero también se puede graficar la respuesta en el tiempo de la función de trasferencia de la planta con una compensación pero sin alinealidad, o al revés, esto resulta muy útil para observar en el gráfico de la respuesta en el tiempo, como cambia esta cuanto a la planta se le agrega ya sea una compensación o una alinealidad, y por último se podría graficar la respuesta en el tiempo para el caso en que estén todos los tres elementos presentes, es decir el sistema completo mostrado en la figura 2.20.

Para el proceso de graficación se debe realizar una discretización de las funciones utilizadas, y esto se lo hace en base a la figura 2.20, en

donde como puede apreciarse se presentan variables a la salida de cada módulo que dependen de la entrada del mismo como también de la característica que presente dicho módulo, lo cual se analiza a continuación con más detalle.

Primeramente la entrada del sistema aparece igual a 1, esto es por que se está considerando que para el tiempo cero hay una transición de la señal de entrada desde cero a uno, es decir se trata de una entrada paso unitaria, pero como el análisis posterior se lo realiza solo para valores de tiempo positivo, entonces se considerará que la entrada del sistema permanece en uno.

Luego aparece una señal de error la misma que es la diferencia entre la entrada y la salida en un instante de tiempo, siendo igual a:

$$u(k) = 1-Y(k)$$

Conocida esta señal de error, la misma pasará a ser la entrada de la función de transferencia de la compensación empleada, entonces para determinar el valor de la salida que presenta dicho compensador en el tiempo se procede a discretizar el compensador tipo PID empleado, encontrándose la siguiente expresión.[1]

$$m = Kp \times u + Ki \times \int u dt + Kd \frac{du}{dt}$$

$$m(s) = \left( Kp + \frac{Ki}{s} + Kd \times s \right) u(s)$$

---

1 PINTO Germánico, Identificación de Sistemas en Tiempo Real, EPN, 1991, PAGS C1-C2

Realizando la transformación Z con aproximaciones de la integral por el método de las sumas de trapecios se obtiene:

$$\frac{m(z)}{u(z)} = Kp + Ki \times \frac{T(z+1)}{2(z-1)} + \frac{Kd}{T} \times \frac{z-1}{z}$$

En donde T = Período de muestreo

$$m(k) = m(k-1) + b1 \times u(k) + b2 \times u(k-1) + b3 \times u(k-2)$$

b1 = Kp + Ki*T/2 + Kd/T

b2 = -Kp + Ki*T/2 - 2*Kd/T

b3 = Kd/T

Con este procedimiento se puede determinar los valores de la salida del compensador m(k) en función de la entrada n(k), pero debe notarse que m(k) es una ecuación de diferencias de segundo orden, entonces depende también de m(k-1), de u(k-1) y u(k-2).

Si se presenta el caso de que no se disponga de una compensación entonces:

$$m(k) = u(k)$$

Luego ya con el valor de m(k), este se convierte en la entrada de la no linealidad, cuya salida es n(k), pero para este caso la determinación de n(k) depende del tipo de alinealidad escogida, y se lo realiza mediante la característica gráfica que presenta la no linealidad.

Ya conociendo el valor de n(k) que depende de la no linealidad, o que es igual a m(k) en caso de no usar alinealidad alguna, se procede a encontrar la salida del sistema Y(k) para la función de transferencia de la planta considerando la entrada a la misma como n(k).

Para el caso de la función de transferencia de la planta se debe considerar el siguiente método de discretización:

$$Y(k) = \underline{C}*\underline{X}(k)$$

$$\underline{X}(k+1) = \underline{A}_D*\underline{X}(k) + \underline{B}_D*n(k)$$

En donde:

$\underline{X}(k)$ es la matriz de los vectores de estado

$\underline{A}_D, \underline{B}_D$ = Matrices discretizadas que dependen de la función de transferencia y el período de discretización empleado.

Para obtener la función de transferencia como matrices discretizadas, se puede emplear la representación canónica controlable o la observable, para el programa se ha empleado la forma canónica controlable.

Una vez representada la función de transferencia como matrices, se obtiene la discretización de la siguiente forma:

$$\underline{A}_D = \underline{I} + \underline{A}*T + \underline{A}^2T^2/2! + \ldots\ldots$$

$$\underline{B}_D = \underline{B}*T$$

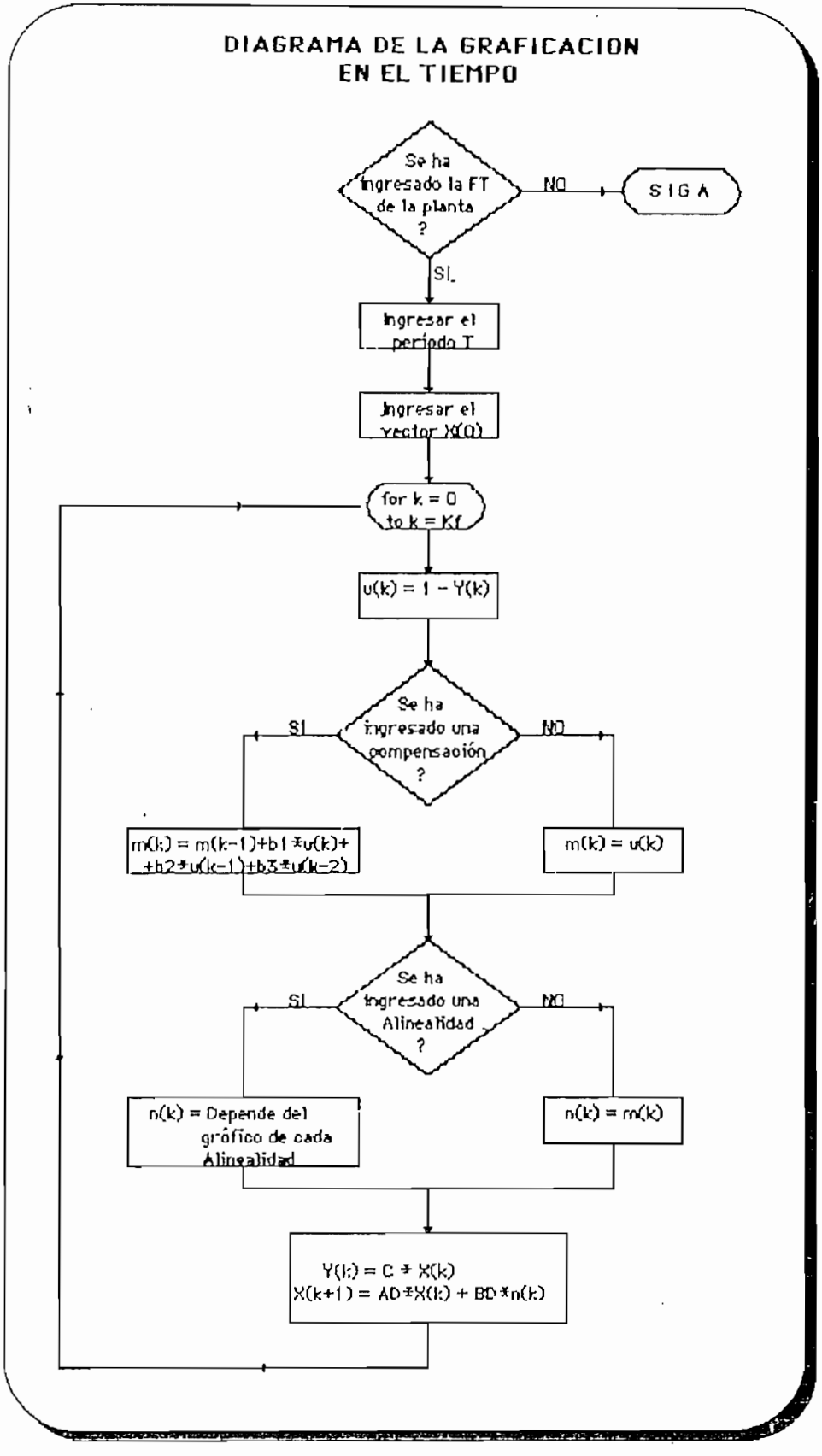



*Fig. 2.21 Diagrama de flujo de la discretización del Sistema*

Al tener discretizado todo el sistema, se puede encontrar el valor de Y(k), pero como se trata de un sistema interactivo, se deben realizar varias veces este proceso, para lo cual se cambia el valor de k, tal como puede apreciarse en el diagrama de flujo de la figura 2.21 en donde se va almacenando los valores de Y(k) obtenidos cada vez, en un vector, para luego en la graficación hacerla punto a punto.

Una cosa importante que se aprecia en la figura 2.21 es que se requiere el ingreso del valor del período de muestreo, el mismo que no puede ser arbitrario, ya que depende de las características de la planta, pero para poder establecer el valor del período óptimo se debe analizar el diagrama del lugar geométrico de las raíces que presentaría la planta.

Usando este diagrama de las raíces se puede determinar si existen raíces imaginarias, en donde la parte del eje de las ordenadas representa la frecuencia $W_1$, la misma que es igual a $W_1 = 2\pi/T_1$, y debe considerarse entonces que el período de muestreo sea menor que este entonces $Tm << T_1$, pero si no hay polos complejos, se analiza directamente con el polo más cercano del eje jw, cuya magnitud representa el inverso de la constante de tiempo $\tau$ que debe ser mucho mayor que el período de muestreo, este método de determinación del período de muestreo se lo aprecia mejor en la figura 2.22

Si no se desea usar este método, el utilizar un período de muestreo grande causaría la perdida de información, obteniéndose una curva de

respuesta en el tiempo errónea, entonces es más bien recomendable el emplear un período de muestreo muy pequeño.

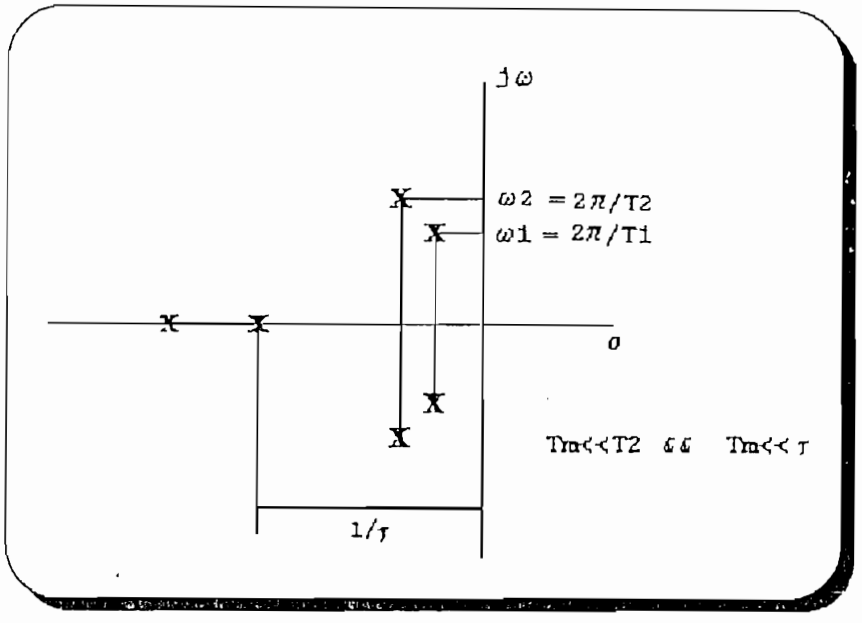



*Fig. 2.22 Análisis de los valores propios en el plano complejo*

Luego de ingresado el período de muestreo debe también ingresarse los valores iniciales del vector de estado, ya que por medio de este vector inicial se estaría indicando las condiciones iniciales para la graficación, y debe considerarse que el ingreso de este vector no puede ser del todo arbitrario, debido a que influye mucho el estar utilizando alinealidades, ya que si se tiene un ciclo límite estable y otro inestable, en donde el uno encierra al anterior en el espacio de estado, puede darse el caso de que el ingreso de las condiciones iniciales del vector de estado pongan al punto inicial en una zona que haga que la salida se vuelva creciente, o tienda a la estabilidad, entonces es necesario graficar las trayectorias de fase para el sistema empleado, y analizarlas las mismas para así encontrar el punto óptimo para el vector de estado inicial, pero se presenta en este caso el problema de que solamente se puede graficar las trayectorias de

fase hasta una función de tercer orden ya que este gráfico correspondería a un diagrama en tres dimensiones, mientras que el graficar en un plano de más de tres dimensiones no se tiene una apreciación física.

Una vez ingresadas las condiciones iniciales para el gráfico se ingresa como se ve en la figura 2.21 a un lazo en donde se varía k, y luego de un número grande de puntos se obtiene un vector que guarda la información de la respuesta en el tiempo, anotando también que cada intervalo en que se repite el lazo, corresponde a un período de tiempo, entonces si el número total de veces que se realiza este cálculo es Kf se tendrá la respuesta en el tiempo desde cero hasta T*Kf, siendo T el período de muestreo.

Una vez que se tiene el gráfico en pantalla, se puede analizar el mismo para encontrar máximo sobreimpulso, tiempo de respuesta entre otros datos que se pueden extraer del gráfico, y con ellos uno puede ver si se debe mejorar el sistema ingresando otra compensación u otra alinealidad, dependiendo de los requerimientos a los que se desee llegar.

En el capítulo siguiente se presentan una serie de ejemplos con los cuales se verá más claramente el funcionamiento del programa.

# CAPITULO III

## EJERCICIOS

# CAPITULO III

## 3.1 EJERCICIOS PARA COMPROBAR RESULTADOS

A continuación se muestran varios ejemplos utilizados para comprobar los resultados que muestran los libros, comparando con los que se obtiene con este programa, para lo cual se realiza un análisis de los resultados obtenidos tanto en forma gráfica como con respecto a las tablas de datos que facilita el programa

*Ejercicio # 1* [1]

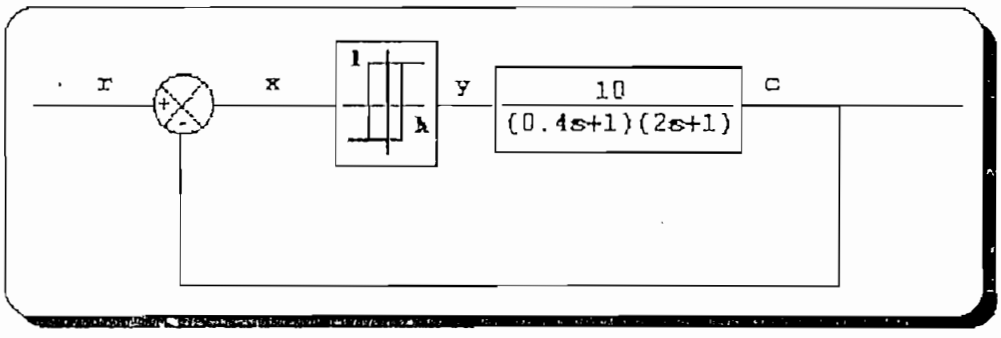



*Fig. 3.1 Sistema de control no lineal*

En la figura anterior aparece un control no lineal a base de una histéresis colocada antes de la planta, la misma que es de segundo orden; si se realiza una simulación en el programa para valores de h=0.1, se puede observar que al graficar las curvas de G(jw) con -1/N, se obtiene un punto

---

1 OGATA KATSUHIKO. INGENIERIA DE CONTROL MODERNA. PRENTICE HALL 1985. pags 598-599

de cruce que corresponde a un ciclo limite estable, con valores de Wc=7.538 y Ac=0.265, como se aprecia en la figura 3.2.

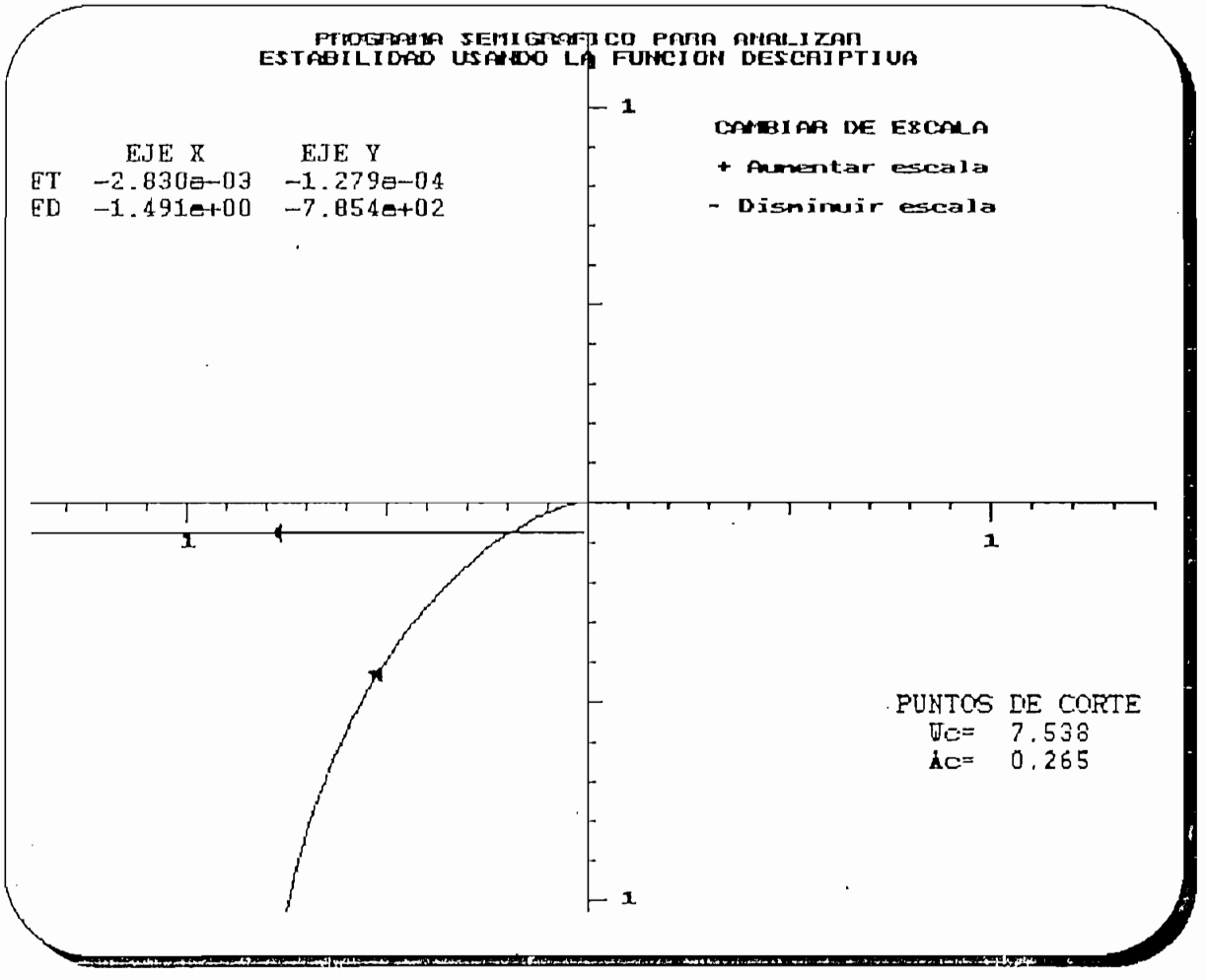



*Fig. 3.2 Diagrama de -1/N y G(jw) para el sistema de la figura 3.1*

Además debe notarse que el lugar de -1/N para este tipo de alinealidad esta formado por rectas paralelas al eje real y que se mueven de derecha a izquierda conforme aumenta el valor de la amplitud A, rectas que se distancian de dicho eje según como se cambie el valor de la histéresis, entonces por ejemplo si se utiliza un valor de h=0.2, la recta del lugar de -1/N aparecerá desplazada más abajo que cuando h=0.1.

Este desplazamiento de la respuesta de -1/N, trae consigo el considerar que en este caso se reducirá la frecuencia del ciclo límite pero su amplitud aumentará, tal es así que para h=0.2 se obtiene que: Wc=5.852 y Ac=0.426

Analizando la curva G(jw) en la figura anterior, se aprecia que la planta inicialmente es estable ya que la curva G(jw) no encierra al punto -1+j0, pero al colocar la alinealidad en el sistema se presenta un ciclo límite estable lo cual ocasiona que la respuesta en el tiempo presente una curva oscilante con frecuencia Wc y amplitud Ac, figura 3.3, pero para llegar a este punto de oscilación le toma un cierto tiempo, ya que originalmente se presenta un sobrepulso, que depende de las condiciones iniciales que se pongan en el vector de estado.

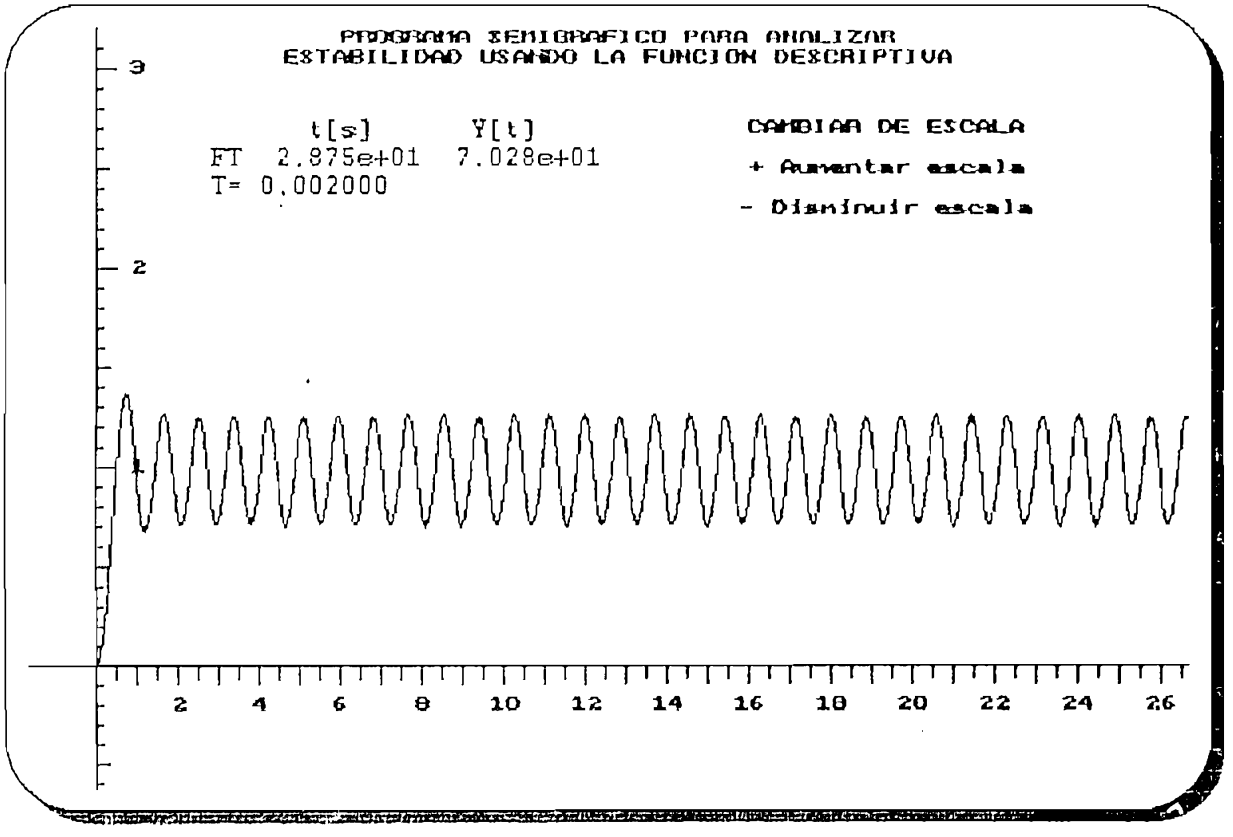



*Fig. 3.3 Respuesta en el tiempo del sistema de la figura 3.1*

La determinación de la forma de crecimiento o desplazamiento de la curva G(jw) o de -I/N, se puede ver en el momento de ejecución del programa, pero para mayor visualización se presenta a continuación la tabla de valores que se obtiene para cada una de estas curvas.

FUNCION DE TRANSFERENCIA

Grado del Numerador- 0
Grado del Denominador- 2

NUMERADOR
a0= 10.000000

DENOMINADOR
b0- 1.000000
b1- 2.400000
b2- 0.800000

RESPUESTA DE FRECUENCIA DE LA FUNCION DE TRANSFERENCIA

| W | Real FT. | Imag. FT. | Módulo FT | Angulo FT |
|---|---|---|---|---|
| 0.05 | 9.877e+00 | -1.188e+00 | 9.948e+00 | 353.144 |
| 0.10 | 9.523e+00 | -2.304e+00 | 9.798e+00 | 346.399 |
| 0.15 | 8.977e+00 | -3.291e+00 | 9.561e+00 | 339.867 |
| 0.20 | 8.292e+00 | -4.112e+00 | 9.255e+00 | 333.625 |
| 0.25 | 7.525e+00 | -4.752e+00 | 8.900e+00 | 327.724 |
| 0.30 | 6.727e+00 | -5.219e+00 | 8.514e+00 | 322.193 |
| 0.35 | 5.937e+00 | -5.529e+00 | 8.113e+00 | 317.038 |
| 0.40 | 5.184e+00 | -5.708e+00 | 7.711e+00 | 312.250 |
| 0.45 | 4.485e+00 | -5.780e+00 | 7.315e+00 | 307.809 |
| 0.50 | 3.846e+00 | -5.769e+00 | 6.934e+00 | 303.690 |
| 0.55 | 3.272e+00 | -5.697e+00 | 6.570e+00 | 299.866 |
| 0.60 | 2.759e+00 | -5.580e+00 | 6.225e+00 | 296.310 |
| 0.65 | 2.305e+00 | -5.432e+00 | 5.901e+00 | 292.994 |
| 0.70 | 1.905e+00 | -5.263e+00 | 5.597e+00 | 289.895 |
| 0.75 | 1.553e+00 | -5.081e+00 | 5.313e+00 | 286.991 |
| 0.80 | 1.243e+00 | -4.892e+00 | 5.048e+00 | 284.261 |
| 0.85 | 9.724e-01 | -4.701e+00 | 4.800e+00 | 281.688 |
| 0.90 | 7.349e-01 | -4.510e+00 | 4.569e+00 | 279.256 |
| 0.95 | 5.269e-01 | -4.322e+00 | 4.354e+00 | 276.952 |
| 1.00 | 3.448e-01 | -4.138e+00 | 4.152e+00 | 274.764 |
| 1.05 | 1.854e-01 | -3.960e+00 | 3.964e+00 | 272.681 |
| 1.10 | 4.591e-02 | -3.787e+00 | 3.788e+00 | 270.694 |
| 1.15 | -7.611e-02 | -3.622e+00 | 3.622e+00 | 268.796 |
| 1.20 | -1.827e-01 | -3.463e+00 | 3.467e+00 | 266.979 |
| 1.25 | -2.759e-01 | -3.310e+00 | 3.322e+00 | 265.236 |
| 1.30 | -3.571e-01 | -3.165e+00 | 3.185e+00 | 263.563 |
| 1.35 | -4.277e-01 | -3.026e+00 | 3.056e+00 | 261.954 |
| 1.40 | -4.891e-01 | -2.894e+00 | 2.935e+00 | 260.405 |
| 1.45 | -5.423e-01 | -2.767e+00 | 2.820e+00 | 258.912 |
| 1.50 | -5.882e-01 | -2.647e+00 | 2.712e+00 | 257.471 |
| 1.55 | -6.277e-01 | -2.533e+00 | 2.609e+00 | 256.080 |
| 1.60 | -6.615e-01 | -2.424e+00 | 2.512e+00 | 254.735 |
| 1.65 | -6.901e-01 | -2.320e+00 | 2.420e+00 | 253.434 |
| 1.70 | -7.143e-01 | -2.221e+00 | 2.333e+00 | 252.174 |
| 1.75 | -7.345e-01 | -2.127e+00 | 2.251e+00 | 250.953 |

| W | Real FT. | Imag. FT. | Módulo FT | Ángulo FT |
|---|---|---|---|---|
| 1.80 | -7.511e-01 | -2.038e+00 | 2.172e+00 | 249.770 |
| 1.85 | -7.645e-01 | -1.953e+00 | 2.097e+00 | 248.623 |
| 1.90 | -7.751e-01 | -1.872e+00 | 2.026e+00 | 247.509 |
| 1.95 | -7.832e-01 | -1.795e+00 | 1.958e+00 | 246.427 |
| 2.00 | -7.891e-01 | -1.722e+00 | 1.894e+00 | 245.376 |
| 2.05 | -7.930e-01 | -1.652e+00 | 1.832e+00 | 244.355 |
| 2.10 | -7.952e-01 | -1.585e+00 | 1.774e+00 | 243.362 |
| 2.15 | -7.958e-01 | -1.522e+00 | 1.717e+00 | 242.396 |
| 2.20 | -7.950e-01 | -1.462e+00 | 1.664e+00 | 241.456 |
| 2.25 | -7.930e-01 | -1.404e+00 | 1.612e+00 | 240.542 |
| 2.30 | -7.899e-01 | -1.349e+00 | 1.563e+00 | 239.651 |
| 2.35 | -7.859e-01 | -1.297e+00 | 1.516e+00 | 238.783 |
| 2.40 | -7.810e-01 | -1.247e+00 | 1.471e+00 | 237.937 |
| 2.45 | -7.754e-01 | -1.199e+00 | 1.428e+00 | 237.113 |
| 2.50 | -7.692e-01 | -1.154e+00 | 1.387e+00 | 236.310 |
| 2.55 | -7.625e-01 | -1.110e+00 | 1.347e+00 | 235.526 |
| 2.60 | -7.552e-01 | -1.069e+00 | 1.309e+00 | 234.762 |
| 2.65 | -7.475e-01 | -1.030e+00 | 1.272e+00 | 234.017 |
| 2.70 | -7.395e-01 | -9.918e-01 | 1.237e+00 | 233.289 |
| 2.75 | -7.312e-01 | -9.557e-01 | 1.203e+00 | 232.579 |
| 2.80 | -7.227e-01 | -9.211e-01 | 1.171e+00 | 231.885 |
| 2.85 | -7.139e-01 | -8.882e-01 | 1.140e+00 | 231.208 |
| 2.90 | -7.050e-01 | -8.566e-01 | 1.109e+00 | 230.546 |
| 2.95 | -6.959e-01 | -8.264e-01 | 1.080e+00 | 229.900 |
| 3.00 | -6.868e-01 | -7.975e-01 | 1.052e+00 | 229.268 |
| 3.05 | -6.775e-01 | -7.699e-01 | 1.026e+00 | 228.650 |
| 3.10 | -6.682e-01 | -7.434e-01 | 9.996e-01 | 228.047 |
| 3.15 | -6.589e-01 | -7.180e-01 | 9.746e-01 | 227.457 |
| 3.20 | -6.496e-01 | -6.937e-01 | 9.504e-01 | 226.879 |
| 3.25 | -6.404e-01 | -6.704e-01 | 9.271e-01 | 226.315 |
| 3.30 | -6.311e-01 | -6.481e-01 | 9.046e-01 | 225.762 |
| 3.35 | -6.219e-01 | -6.267e-01 | 8.829e-01 | 225.222 |
| 3.40 | -6.127e-01 | -6.062e-01 | 8.619e-01 | 224.693 |
| 3.45 | -6.036e-01 | -5.865e-01 | 8.416e-01 | 224.175 |
| 3.50 | -5.946e-01 | -5.676e-01 | 8.220e-01 | 223.668 |
| 3.55 | -5.857e-01 | -5.494e-01 | 8.030e-01 | 223.171 |
| 3.60 | -5.768e-01 | -5.320e-01 | 7.847e-01 | 222.685 |
| 3.65 | -5.681e-01 | -5.152e-01 | 7.669e-01 | 222.209 |
| 3.70 | -5.594e-01 | -4.992e-01 | 7.497e-01 | 221.742 |
| 3.75 | -5.509e-01 | -4.837e-01 | 7.331e-01 | 221.285 |
| 3.80 | -5.425e-01 | -4.688e-01 | 7.170e-01 | 220.837 |
| 3.85 | -5.342e-01 | -4.546e-01 | 7.014e-01 | 220.397 |
| 3.90 | -5.260e-01 | -4.408e-01 | 6.863e-01 | 219.967 |
| 3.95 | -5.179e-01 | -4.276e-01 | 6.716e-01 | 219.544 |
| 4.00 | -5.099e-01 | -4.149e-01 | 6.574e-01 | 219.130 |
| 4.40 | -4.508e-01 | -3.285e-01 | 5.578e-01 | 216.088 |
| 4.80 | -3.993e-01 | -2.639e-01 | 4.786e-01 | 213.459 |
| 5.20 | -3.548e-01 | -2.146e-01 | 4.147e-01 | 211.169 |
| 5.60 | -3.166e-01 | -1.766e-01 | 3.625e-01 | 209.160 |
| 6.00 | -2.836e-01 | -1.469e-01 | 3.194e-01 | 207.384 |
| 6.40 | -2.551e-01 | -1.234e-01 | 2.834e-01 | 205.804 |
| 6.80 | -2.305e-01 | -1.045e-01 | 2.530e-01 | 204.391 |
| 7.20 | -2.090e-01 | -8.923e-02 | 2.272e-01 | 203.121 |
| 7.60 | -1.902e-01 | -7.675e-02 | 2.051e-01 | 201.973 |
| 8.00 | -1.738e-01 | -6.647e-02 | 1.861e-01 | 200.930 |
| 8.40 | -1.593e-01 | -5.792e-02 | 1.695e-01 | 199.980 |
| 8.80 | -1.465e-01 | -5.075e-02 | 1.550e-01 | 199.111 |
| 9.20 | -1.351e-01 | -4.471e-02 | 1.423e-01 | 198.313 |

| W | Real FT. | Imag. FT. | Módulo FT | Ángulo FT |
|---|---|---|---|---|
| 10.00 | -1.159e-01 | -3.521e-02 | 1.211e-01 | 196.899 |
| 10.40 | -1.077e-01 | -3.144e-02 | 1.122e-01 | 196.269 |
| 10.80 | -1.004e-01 | -2.819e-02 | 1.043e-01 | 195.684 |
| 11.20 | -9.379e-02 | -2.537e-02 | 9.716e-02 | 195.139 |
| 11.60 | -8.778e-02 | -2.292e-02 | 9.073e-02 | 194.630 |
| 12.00 | -8.233e-02 | -2.076e-02 | 8.491e-02 | 194.154 |
| 12.40 | -7.736e-02 | -1.887e-02 | 7.963e-02 | 193.708 |
| 12.80 | -7.282e-02 | -1.720e-02 | 7.482e-02 | 193.288 |
| 13.20 | -6.866e-02 | -1.572e-02 | 7.044e-02 | 192.894 |
| 13.60 | -6.484e-02 | -1.440e-02 | 6.642e-02 | 192.522 |
| 14.00 | -6.133e-02 | -1.323e-02 | 6.274e-02 | 192.170 |
| 14.40 | -5.810e-02 | -1.218e-02 | 5.936e-02 | 191.838 |
| 14.80 | -5.510e-02 | -1.123e-02 | 5.624e-02 | 191.523 |
| 15.20 | -5.234e-02 | -1.039e-02 | 5.336e-02 | 191.224 |
| 15.60 | -4.977e-02 | -9.621e-03 | 5.069e-02 | 190.940 |
| 16.00 | -4.739e-02 | -8.928e-03 | 4.822e-02 | 190.671 |
| 16.40 | -4.517e-02 | -8.301e-03 | 4.592e-02 | 190.414 |
| 16.80 | -4.310e-02 | -7.730e-03 | 4.379e-02 | 190.169 |
| 17.20 | -4.117e-02 | -7.211e-03 | 4.180e-02 | 189.935 |
| 17.60 | -3.936e-02 | -6.737e-03 | 3.994e-02 | 189.712 |
| 18.00 | -3.768e-02 | -6.303e-03 | 3.820e-02 | 189.498 |
| 18.40 | -3.609e-02 | -5.906e-03 | 3.657e-02 | 189.294 |
| 18.80 | -3.460e-02 | -5.542e-03 | 3.505e-02 | 189.098 |
| 19.20 | -3.321e-02 | -5.206e-03 | 3.361e-02 | 188.910 |
| 19.60 | -3.189e-02 | -4.897e-03 | 3.227e-02 | 188.730 |
| 20.00 | -3.065e-02 | -4.613e-03 | 3.100e-02 | 188.557 |
| 20.40 | -2.949e-02 | -4.349e-03 | 2.980e-02 | 188.391 |
| 20.80 | -2.838e-02 | -4.105e-03 | 2.868e-02 | 188.231 |
| 21.20 | -2.734e-02 | -3.880e-03 | 2.761e-02 | 188.077 |
| 21.60 | -2.635e-02 | -3.670e-03 | 2.661e-02 | 187.928 |
| 22.00 | -2.542e-02 | -3.475e-03 | 2.565e-02 | 187.785 |
| 22.40 | -2.453e-02 | -3.294e-03 | 2.475e-02 | 187.647 |
| 22.80 | -2.369e-02 | -3.125e-03 | 2.390e-02 | 187.514 |
| 23.20 | -2.289e-02 | -2.967e-03 | 2.308e-02 | 187.385 |
| 23.60 | -2.213e-02 | -2.820e-03 | 2.231e-02 | 187.261 |
| 24.00 | -2.141e-02 | -2.682e-03 | 2.158e-02 | 187.140 |
| 24.40 | -2.073e-02 | -2.554e-03 | 2.088e-02 | 187.024 |
| 24.80 | -2.007e-02 | -2.433e-03 | 2.022e-02 | 186.911 |
| 25.20 | -1.945e-02 | -2.320e-03 | 1.958e-02 | 186.802 |
| 25.60 | -1.885e-02 | -2.213e-03 | 1.898e-02 | 186.697 |
| 26.00 | -1.828e-02 | -2.113e-03 | 1.840e-02 | 186.594 |
| 26.40 | -1.774e-02 | -2.019e-03 | 1.785e-02 | 186.495 |
| 26.80 | -1.722e-02 | -1.931e-03 | 1.733e-02 | 186.398 |
| 27.20 | -1.672e-02 | -1.847e-03 | 1.682e-02 | 186.305 |
| 27.60 | -1.624e-02 | -1.769e-03 | 1.634e-02 | 186.214 |
| 28.00 | -1.579e-02 | -1.694e-03 | 1.588e-02 | 186.125 |
| 28.40 | -1.535e-02 | -1.624e-03 | 1.544e-02 | 186.039 |
| 28.80 | -1.493e-02 | -1.558e-03 | 1.501e-02 | 185.956 |
| 29.20 | -1.453e-02 | -1.495e-03 | 1.460e-02 | 185.875 |
| 29.60 | -1.414e-02 | -1.435e-03 | 1.421e-02 | 185.795 |
| 30.00 | -1.377e-02 | -1.379e-03 | 1.384e-02 | 185.718 |
| 30.40 | -1.341e-02 | -1.325e-03 | 1.348e-02 | 185.644 |
| 30.80 | -1.307e-02 | -1.275e-03 | 1.313e-02 | 185.571 |
| 31.20 | -1.274e-02 | -1.227e-03 | 1.280e-02 | 185.499 |
| 31.60 | -1.242e-02 | -1.181e-03 | 1.248e-02 | 185.430 |
| 32.00 | -1.212e-02 | -1.137e-03 | 1.217e-02 | 185.362 |
| 32.40 | -1.182e-02 | -1.096e-03 | 1.187e-02 | 185.296 |
| 32.80 | -1.154e-02 | -1.056e-03 | 1.158e-02 | 185.232 |

| W | Real FT. | Imag. FT. | Módulo FT | Ángulo FT |
|---|---|---|---|---|
| 33.60 | -1.100e-02 | -9.829e-04 | 1.104e-02 | 185.108 |
| 34.00 | -1.074e-02 | -9.488e-04 | 1.078e-02 | 185.048 |
| 34.40 | -1.049e-02 | -9.162e-04 | 1.053e-02 | 184.989 |
| 34.80 | -1.026e-02 | -8.851e-04 | 1.029e-02 | 184.932 |
| 35.20 | -1.003e-02 | -8.553e-04 | 1.006e-02 | 184.876 |
| 35.60 | -9.803e-03 | -8.269e-04 | 9.838e-03 | 184.822 |
| 36.00 | -9.588e-03 | -7.997e-04 | 9.621e-03 | 184.768 |
| 36.40 | -9.379e-03 | -7.738e-04 | 9.411e-03 | 184.716 |
| 36.80 | -9.178e-03 | -7.489e-04 | 9.208e-03 | 184.665 |
| 37.20 | -8.982e-03 | -7.250e-04 | 9.012e-03 | 184.615 |
| 37.60 | -8.793e-03 | -7.022e-04 | 8.821e-03 | 184.566 |
| 38.00 | -8.610e-03 | -6.803e-04 | 8.637e-03 | 184.518 |
| 38.40 | -8.433e-03 | -6.594e-04 | 8.458e-03 | 184.471 |
| 38.80 | -8.261e-03 | -6.392e-04 | 8.285e-03 | 184.425 |
| 39.20 | -8.094e-03 | -6.199e-04 | 8.117e-03 | 184.380 |
| 39.60 | -7.932e-03 | -6.014e-04 | 7.955e-03 | 184.336 |
| 40.00 | -7.775e-03 | -5.836e-04 | 7.797e-03 | 184.292 |
| 40.40 | -7.622e-03 | -5.664e-04 | 7.643e-03 | 184.250 |
| 40.80 | -7.474e-03 | -5.500e-04 | 7.495e-03 | 184.209 |
| 41.20 | -7.331e-03 | -5.342e-04 | 7.350e-03 | 184.168 |
| 41.60 | -7.191e-03 | -5.189e-04 | 7.210e-03 | 184.128 |
| 42.00 | -7.055e-03 | -5.043e-04 | 7.073e-03 | 184.089 |
| 42.40 | -6.923e-03 | -4.902e-04 | 6.941e-03 | 184.050 |
| 42.80 | -6.795e-03 | -4.766e-04 | 6.812e-03 | 184.012 |
| 43.20 | -6.670e-03 | -4.635e-04 | 6.686e-03 | 183.975 |
| 43.60 | -6.549e-03 | -4.509e-04 | 6.564e-03 | 183.939 |
| 44.00 | -6.431e-03 | -4.388e-04 | 6.446e-03 | 183.903 |
| 44.40 | -6.316e-03 | -4.270e-04 | 6.330e-03 | 183.868 |
| 44.80 | -6.204e-03 | -4.157e-04 | 6.218e-03 | 183.833 |
| 45.20 | -6.095e-03 | -4.048e-04 | 6.109e-03 | 183.800 |
| 45.60 | -5.989e-03 | -3.943e-04 | 6.002e-03 | 183.766 |
| 46.00 | -5.886e-03 | -3.841e-04 | 5.898e-03 | 183.734 |
| 46.40 | -5.785e-03 | -3.743e-04 | 5.797e-03 | 183.701 |
| 46.80 | -5.687e-03 | -3.648e-04 | 5.699e-03 | 183.670 |
| 47.20 | -5.591e-03 | -3.556e-04 | 5.603e-03 | 183.639 |
| 47.60 | -5.498e-03 | -3.467e-04 | 5.509e-03 | 183.608 |
| 48.00 | -5.407e-03 | -3.381e-04 | 5.418e-03 | 183.578 |
| 48.40 | -5.318e-03 | -3.298e-04 | 5.329e-03 | 183.549 |
| 50.40 | -4.906e-03 | -2.922e-04 | 4.915e-03 | 183.408 |
| 52.40 | -4.540e-03 | -2.600e-04 | 4.547e-03 | 183.278 |
| 54.40 | -4.213e-03 | -2.324e-04 | 4.219e-03 | 183.158 |
| 56.40 | -3.920e-03 | -2.086e-04 | 3.926e-03 | 183.046 |
| 58.40 | -3.657e-03 | -1.879e-04 | 3.662e-03 | 182.942 |
| 60.40 | -3.419e-03 | -1.699e-04 | 3.423e-03 | 182.844 |
| 62.40 | -3.204e-03 | -1.541e-04 | 3.208e-03 | 182.753 |
| 64.40 | -3.008e-03 | -1.402e-04 | 3.012e-03 | 182.668 |
| 66.40 | -2.830e-03 | -1.279e-04 | 2.833e-03 | 182.588 |

FUNCION DESCRIPTIVA
HISTERESIS IDEAL
U - 1.000000
D - 0.100000

RESPUESTA DE FRECUENCIA DE LA FUNCION DESCRIPTIVA

| A | Real FD. | Imag. FD. | Módulo FD | Angulo FD |
|---|---|---|---|---|
| 0.101 | -1.113e-02 | -7.854e-02 | 7.933e-02 | 261.931 |
| 0.401 | -3.050e-01 | -7.854e-02 | 3.149e-01 | 194.441 |
| 0.701 | -5.449e-01 | -7.854e-02 | 5.506e-01 | 188.201 |
| 1.001 | -7.823e-01 | -7.854e-02 | 7.862e-01 | 185.733 |
| 1.301 | -1.019e+00 | -7.854e-02 | 1.022e+00 | 184.408 |
| 1.601 | -1.255e+00 | -7.854e-02 | 1.257e+00 | 183.581 |
| 1.901 | -1.491e+00 | -7.854e-02 | 1.493e+00 | 183.015 |
| 2.201 | -1.727e+00 | -7.854e-02 | 1.729e+00 | 182.604 |
| 2.501 | -1.963e+00 | -7.854e-02 | 1.964e+00 | 182.292 |
| 2.801 | -2.198e+00 | -7.854e-02 | 2.200e+00 | 182.046 |
| 3.101 | -2.434e+00 | -7.854e-02 | 2.436e+00 | 181.848 |
| 3.401 | -2.670e+00 | -7.854e-02 | 2.671e+00 | 181.685 |
| 3.701 | -2.906e+00 | -7.854e-02 | 2.907e+00 | 181.548 |
| 4.001 | -3.141e+00 | -7.854e-02 | 3.142e+00 | 181.432 |
| 4.301 | -3.377e+00 | -7.854e-02 | 3.378e+00 | 181.332 |
| 4.601 | -3.613e+00 | -7.854e-02 | 3.614e+00 | 181.245 |
| 4.901 | -3.848e+00 | -7.854e-02 | 3.849e+00 | 181.169 |
| 5.201 | -4.084e+00 | -7.854e-02 | 4.085e+00 | 181.102 |
| 5.501 | -4.320e+00 | -7.854e-02 | 4.320e+00 | 181.042 |
| 5.801 | -4.555e+00 | -7.854e-02 | 4.556e+00 | 180.988 |
| 6.101 | -4.791e+00 | -7.854e-02 | 4.792e+00 | 180.939 |
| 8.101 | -6.362e+00 | -7.854e-02 | 6.363e+00 | 180.707 |
| 10.101 | -7.933e+00 | -7.854e-02 | 7.933e+00 | 180.567 |
| 12.101 | -9.504e+00 | -7.854e-02 | 9.504e+00 | 180.473 |
| 14.101 | -1.107e+01 | -7.854e-02 | 1.107e+01 | 180.406 |
| 16.101 | -1.265e+01 | -7.854e-02 | 1.265e+01 | 180.356 |
| 18.101 | -1.422e+01 | -7.854e-02 | 1.422e+01 | 180.317 |
| 20.101 | -1.579e+01 | -7.854e-02 | 1.579e+01 | 180.285 |
| 22.101 | -1.736e+01 | -7.854e-02 | 1.736e+01 | 180.259 |
| 24.101 | -1.893e+01 | -7.854e-02 | 1.893e+01 | 180.238 |
| 26.101 | -2.050e+01 | -7.854e-02 | 2.050e+01 | 180.220 |
| 28.101 | -2.207e+01 | -7.854e-02 | 2.207e+01 | 180.204 |
| 30.101 | -2.364e+01 | -7.854e-02 | 2.364e+01 | 180.190 |
| 32.101 | -2.521e+01 | -7.854e-02 | 2.521e+01 | 180.178 |
| 34.101 | -2.678e+01 | -7.854e-02 | 2.678e+01 | 180.168 |
| 36.101 | -2.835e+01 | -7.854e-02 | 2.835e+01 | 180.159 |
| 38.101 | -2.992e+01 | -7.854e-02 | 2.992e+01 | 180.150 |
| 40.101 | -3.150e+01 | -7.854e-02 | 3.150e+01 | 180.143 |
| 42.101 | -3.307e+01 | -7.854e-02 | 3.307e+01 | 180.136 |
| 44.101 | -3.464e+01 | -7.854e-02 | 3.464e+01 | 180.130 |
| 46.101 | -3.621e+01 | -7.854e-02 | 3.621e+01 | 180.124 |
| 48.101 | -3.778e+01 | -7.854e-02 | 3.778e+01 | 180.119 |
| 50.101 | -3.935e+01 | -7.854e-02 | 3.935e+01 | 180.114 |
| 52.101 | -4.092e+01 | -7.854e-02 | 4.092e+01 | 180.110 |
| 54.101 | -4.249e+01 | -7.854e-02 | 4.249e+01 | 180.106 |
| 56.101 | -4.406e+01 | -7.854e-02 | 4.406e+01 | 180.102 |
| 58.101 | -4.563e+01 | -7.854e-02 | 4.563e+01 | 180.099 |
| 60.101 | -4.720e+01 | -7.854e-02 | 4.720e+01 | 180.095 |
| 62.101 | -4.877e+01 | -7.854e-02 | 4.877e+01 | 180.092 |
| 64.101 | -5.034e+01 | -7.854e-02 | 5.034e+01 | 180.089 |
| 66.101 | -5.192e+01 | -7.854e-02 | 5.192e+01 | 180.087 |
| 68.101 | -5.349e+01 | -7.854e-02 | 5.349e+01 | 180.084 |
| 70.101 | -5.506e+01 | -7.854e-02 | 5.506e+01 | 180.082 |
| 72.101 | -5.663e+01 | -7.854e-02 | 5.663e+01 | 180.079 |
| 74.101 | -5.820e+01 | -7.854e-02 | 5.820e+01 | 180.077 |
| 76.101 | -5.977e+01 | -7.854e-02 | 5.977e+01 | 180.075 |
| 78.101 | -6.134e+01 | -7.854e-02 | 6.134e+01 | 180.073 |
| 80.101 | -6.291e+01 | -7.854e-02 | 6.291e+01 | 180.072 |

*Ejercicio # 2* [1]

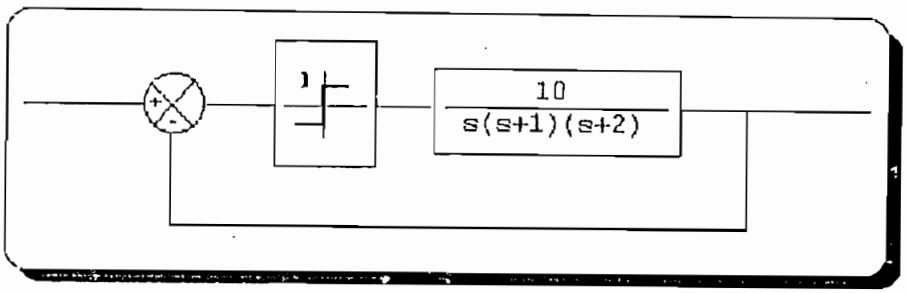



*Fig. 3.4 Sistema de control no lineal con relé ideal*

Para el sistema de la figura 3.4 se obtiene la siguiente respuesta de frecuencia.

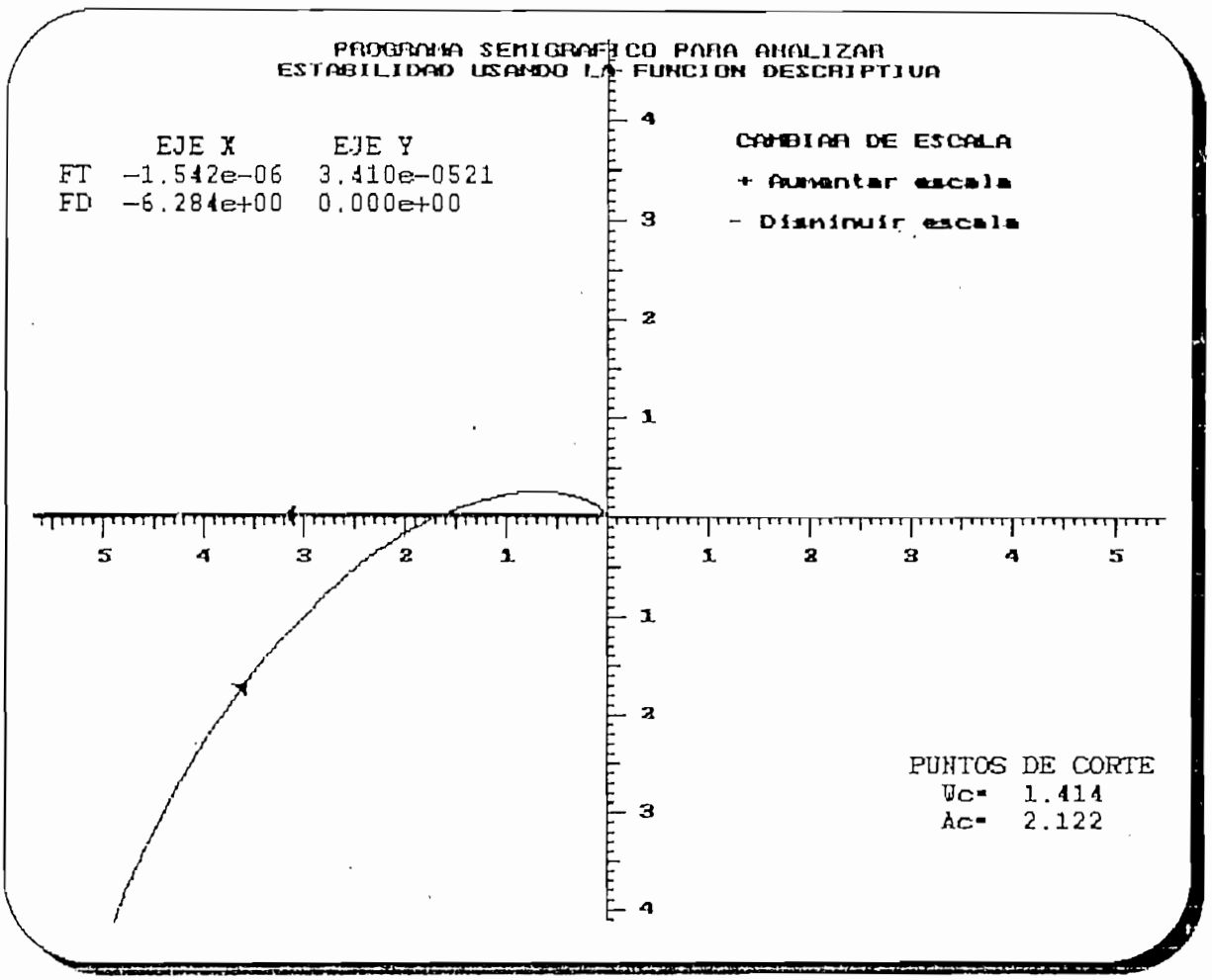



*Fig. 3.5 Gráfico de G(jw) y -1/N para el caso de la figura 3.4*

1 OGATA KATSUHIKO, INGENIERIA DE CONTROL MODERNA, PRENTICE HALL 1985, pags 610

En la figura anterior se observa un punto de cruce que corresponde a un ciclo límite estable de frecuencia Wc=1.414 y amplitud Ac=2.122

Es estable el ciclo límite ya que el desplazamiento de la curva -1/N se da desde la derecha hacia la izquierda conforme aumenta el valor de la amplitud A.

Por lo tanto si se grafica en el tiempo este sistema se obtendrá una respuesta similar a la de la figura 3.3, pero la amplitud de la oscilación mantenida estará en 2.122, al rededor del valor de 1, esto se da por que se considera que la entrada del sistema es paso unitaria; dicha figura se muestra a continuación.

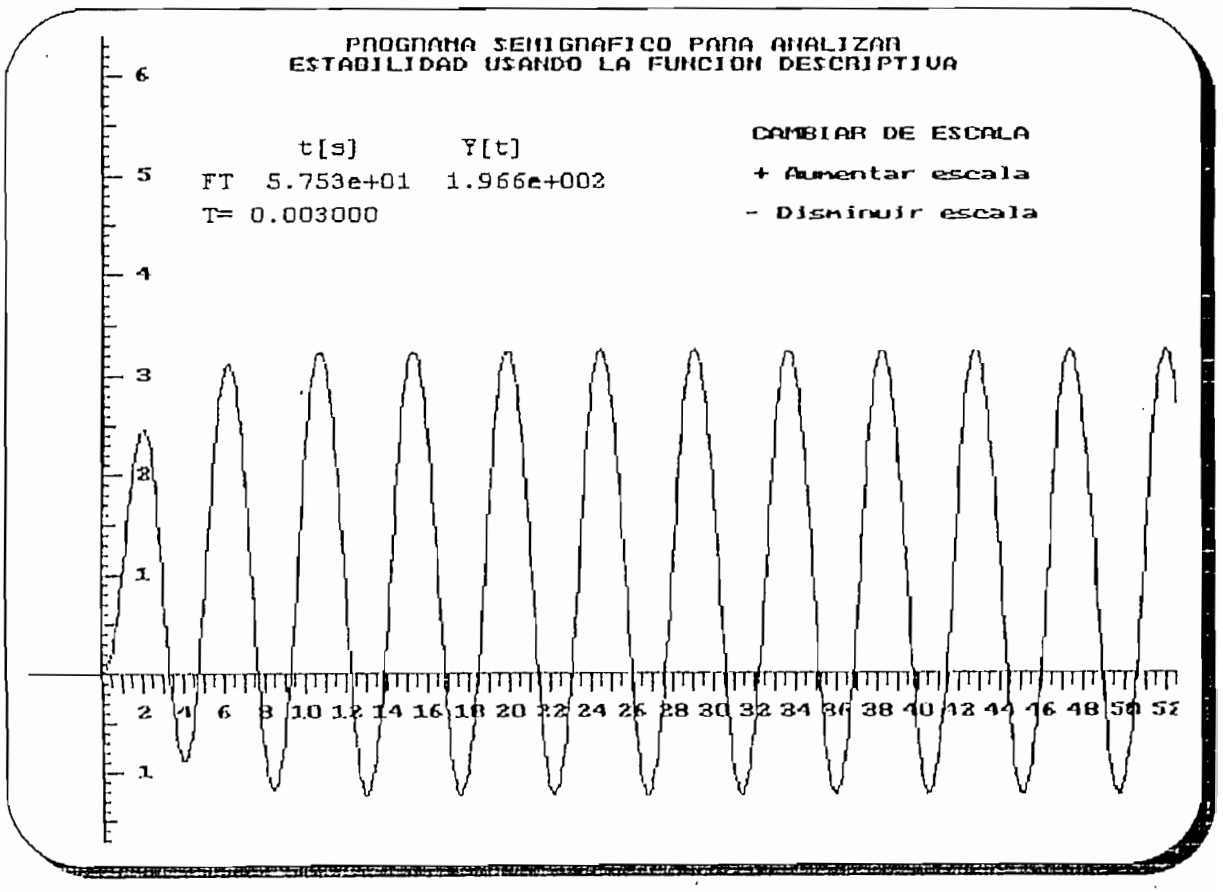



*Fig. 3.5(b) Respuesta en el tiempo para el sistema de la figura 3.4*

Para este problema se presenta a continuación la tabla de valores pero de la respuesta en el tiempo, que es posiblemente donde se puede observar mejor la frecuencia y amplitud en las cuales se mantiene oscilando este sistema.

FUNCION DE TRANSFERENCIA

Grado del Numerador- 0
Grado del Denominador- 3

NUMERADOR
a0- 10.000000

DENOMINADOR
b0- 0.000000
b1- 2.000000
b2- 3.000000
b3- 1.000000

FUNCION DESCRIPTIVA
RELE IDEAL
U - 1.000000

RESPUESTA EN EL TIEMPO

| t[s] | Yk |
|---|---|
| 0.0000 | 0.000e+00 |
| 0.0750 | 5.214e-04 |
| 0.1500 | 4.491e-03 |
| 0.2250 | 1.496e-02 |
| 0.3000 | 3.431e-02 |
| 0.3750 | 6.433e-02 |
| 0.4500 | 1.064e-01 |
| 0.5250 | 1.613e-01 |
| 0.6000 | 2.299e-01 |
| 0.6750 | 3.123e-01 |
| 0.7500 | 4.089e-01 |
| 0.8250 | 5.195e-01 |
| 0.9000 | 6.439e-01 |
| 0.9750 | 7.820e-01 |
| 1.0500 | 9.333e-01 |
| 1.1250 | 1.097e+00 |
| 1.2000 | 1.270e+00 |
| 1.2750 | 1.444e+00 |
| 1.3500 | 1.613e+00 |
| 1.4250 | 1.774e+00 |
| 1.5000 | 1.921e+00 |
| 1.5750 | 2.054e+00 |
| 1.6500 | 2.169e+00 |
| 1.7250 | 2.265e+00 |
| 1.8000 | 2.341e+00 |
| 1.8750 | 2.397e+00 |
| 1.9500 | 2.433e+00 |
| 2.0250 | 2.447e+00 |
| 2.1000 | 2.442e+00 |
| 2.1750 | 2.416e+00 |
| 2.2500 | 2.371e+00 |
| 2.3250 | 2.307e+00 |
| 2.4000 | 2.225e+00 |
| 2.4750 | 2.126e+00 |
| 2.5500 | 2.009e+00 |
| 2.6250 | 1.877e+00 |
| 2.7000 | 1.729e+00 |
| 2.7750 | 1.567e+00 |
| 2.8500 | 1.392e+00 |
| 2.9250 | 1.203e+00 |
| 3.0000 | 1.002e+00 |
| 3.0750 | 7.908e-01 |
| 3.1500 | 5.752e-01 |
| 3.2250 | 3.623e-01 |
| 3.3000 | 1.573e-01 |
| 3.3750 | -3.532e-02 |
| 3.4500 | -2.125e-01 |
| 3.5250 | -3.716e-01 |
| 3.6000 | -5.110e-01 |
| 3.6750 | -6.295e-01 |
| 3.7500 | -7.262e-01 |
| 3.8250 | -8.009e-01 |
| 3.9000 | -8.533e-01 |
| 3.9750 | -8.838e-01 |
| 4.0500 | -8.926e-01 |
| 4.1250 | -8.802e-01 |
| 4.2000 | -8.472e-01 |
| 4.2750 | -7.944e-01 |
| 4.3500 | -7.224e-01 |
| 4.4250 | -6.322e-01 |
| 4.5000 | -5.244e-01 |
| 4.5750 | -4.000e-01 |
| 4.6500 | -2.598e-01 |
| 4.7250 | -1.045e-01 |
| 4.8000 | 6.495e-02 |
| 4.8750 | 2.478e-01 |
| 4.9500 | 4.433e-01 |
| 5.0250 | 6.507e-01 |
| 5.1000 | 8.693e-01 |
| 5.1750 | 1.098e+00 |
| 5.2500 | 1.334e+00 |
| 5.3250 | 1.570e+00 |
| 5.4000 | 1.800e+00 |
| 5.4750 | 2.018e+00 |
| 5.5500 | 2.221e+00 |
| 5.6250 | 2.406e+00 |
| 5.7000 | 2.572e+00 |
| 5.7750 | 2.716e+00 |
| 5.8500 | 2.838e+00 |
| 5.9250 | 2.938e+00 |
| 6.0000 | 3.014e+00 |
| 6.0750 | 3.067e+00 |
| 6.1500 | 3.098e+00 |
| 6.2250 | 3.107e+00 |
| 6.3000 | 3.094e+00 |
| 6.3750 | 3.060e+00 |
| 6.4500 | 3.006e+00 |
| 6.5250 | 2.933e+00 |
| 6.6000 | 2.842e+00 |
| 6.6750 | 2.732e+00 |
| 6.7500 | 2.607e+00 |
| 6.8250 | 2.465e+00 |
| 6.9000 | 2.308e+00 |
| 6.9750 | 2.137e+00 |
| 7.0500 | 1.953e+00 |
| 7.1250 | 1.756e+00 |
| 7.2000 | 1.547e+00 |
| 7.2750 | 1.327e+00 |
| 7.3500 | 1.097e+00 |
| 7.4250 | 8.568e-01 |
| 7.5000 | 6.115e-01 |
| 7.5750 | 3.680e-01 |
| 7.6500 | 1.323e-01 |
| 7.7250 | -9.098e-02 |
| 7.8000 | -2.984e-01 |
| 7.8750 | -4.872e-01 |
| 7.9500 | -6.555e-01 |
| 8.0250 | -8.019e-01 |
| 8.1000 | -9.256e-01 |
| 8.1750 | -1.026e+00 |
| 8.2500 | -1.103e+00 |
| 8.3250 | -1.157e+00 |
| 8.4000 | -1.188e+00 |
| 8.4750 | -1.197e+00 |
| 8.5500 | -1.184e+00 |
| 8.6250 | -1.151e+00 |
| 8.7000 | -1.097e+00 |
| 8.7750 | -1.023e+00 |
| 8.8500 | -9.315e-01 |
| 8.9250 | -8.220e-01 |
| 9.0000 | -6.958e-01 |
| 9.0750 | -5.537e-01 |
| 9.1500 | -3.966e-01 |
| 9.2250 | -2.253e-01 |
| 9.3000 | -4.055e-02 |
| 9.3750 | 1.568e-01 |
| 9.4500 | 3.659e-01 |
| 9.5250 | 5.862e-01 |
| 9.6000 | 8.170e-01 |
| 9.6750 | 1.058e+00 |
| 9.7500 | 1.305e+00 |
| 9.8250 | 1.554e+00 |
| 9.9000 | 1.796e+00 |
| 9.9750 | 2.027e+00 |
| 10.0500 | 2.243e+00 |
| 10.1250 | 2.442e+00 |
| 10.2000 | 2.620e+00 |
| 10.2750 | 2.777e+00 |
| 10.3500 | 2.911e+00 |
| 10.4250 | 3.021e+00 |
| 10.5000 | 3.109e+00 |
| 10.5750 | 3.173e+00 |
| 10.6500 | 3.214e+00 |
| 10.7250 | 3.232e+00 |
| 10.8000 | 3.228e+00 |

| t[s] | Yk |
|---|---|
| 10.8750 | 3.203e+00 |
| 10.9500 | 3.158e+00 |
| 11.0250 | 3.092e+00 |
| 11.1000 | 3.008e+00 |
| 11.1750 | 2.906e+00 |
| 11.2500 | 2.786e+00 |
| 11.3250 | 2.651e+00 |
| 11.4000 | 2.500e+00 |
| 11.4750 | 2.334e+00 |
| 11.5500 | 2.155e+00 |
| 11.6250 | 1.963e+00 |
| 11.7000 | 1.758e+00 |
| 11.7750 | 1.542e+00 |
| 11.8500 | 1.316e+00 |
| 11.9250 | 1.079e+00 |
| 12.0000 | 8.333e-01 |
| 12.0750 | 5.831e-01 |
| 12.1500 | 3.356e-01 |
| 12.2250 | 9.649e-02 |
| 12.3000 | -1.296e-01 |
| 12.3750 | -3.393e-01 |
| 12.4500 | -5.300e-01 |
| 12.5250 | -6.998e-01 |
| 12.6000 | -8.474e-01 |
| 12.6750 | -9.721e-01 |
| 12.7500 | -1.073e+00 |
| 12.8250 | -1.151e+00 |
| 12.9000 | -1.206e+00 |
| 12.9750 | -1.237e+00 |
| 13.0500 | -1.246e+00 |
| 13.1250 | -1.233e+00 |
| 13.2000 | -1.200e+00 |
| 13.2750 | -1.146e+00 |
| 13.3500 | -1.072e+00 |
| 13.4250 | -9.804e-01 |
| 13.5000 | -8.708e-01 |
| 13.5750 | -7.445e-01 |
| 13.6500 | -6.023e-01 |
| 13.7250 | -4.450e-01 |
| 13.8000 | -2.736e-01 |
| 13.8750 | -8.870e-02 |
| 13.9500 | 1.088e-01 |
| 14.0250 | 3.181e-01 |
| 14.1000 | 5.385e-01 |
| 14.1750 | 7.695e-01 |
| 14.2500 | 1.010e+00 |
| 14.3250 | 1.259e+00 |
| 14.4000 | 1.509e+00 |
| 14.4750 | 1.754e+00 |
| 14.5500 | 1.988e+00 |
| 14.6250 | 2.209e+00 |
| 14.7000 | 2.411e+00 |
| 14.7750 | 2.594e+00 |
| 14.8500 | 2.755e+00 |
| 14.9250 | 2.893e+00 |
| 15.0000 | 3.009e+00 |
| 15.0750 | 3.101e+00 |
| 15.3000 | 3.237e+00 |
| 15.3750 | 3.238e+00 |
| 15.4500 | 3.216e+00 |
| 15.5250 | 3.175e+00 |
| 15.6000 | 3.113e+00 |
| 15.6750 | 3.032e+00 |
| 15.7500 | 2.933e+00 |
| 15.8250 | 2.817e+00 |
| 15.9000 | 2.684e+00 |
| 15.9750 | 2.536e+00 |
| 16.0500 | 2.373e+00 |
| 16.1250 | 2.196e+00 |
| 16.2000 | 2.006e+00 |
| 16.2750 | 1.803e+00 |
| 16.3500 | 1.590e+00 |
| 16.4250 | 1.365e+00 |
| 16.5000 | 1.130e+00 |
| 16.5750 | 8.857e-01 |
| 16.6500 | 6.356e-01 |
| 16.7250 | 3.868e-01 |
| 16.8000 | 1.455e-01 |
| 16.8750 | -8.362e-02 |
| 16.9500 | -2.970e-01 |
| 17.0250 | -4.918e-01 |
| 17.1000 | -6.660e-01 |
| 17.1750 | -8.182e-01 |
| 17.2500 | -9.476e-01 |
| 17.3250 | -1.054e+00 |
| 17.4000 | -1.136e+00 |
| 17.4750 | -1.195e+00 |
| 17.5500 | -1.231e+00 |
| 17.6250 | -1.245e+00 |
| 17.7000 | -1.237e+00 |
| 17.7750 | -1.207e+00 |
| 17.8500 | -1.157e+00 |
| 17.9250 | -1.088e+00 |
| 18.0000 | -9.995e-01 |
| 18.0750 | -8.935e-01 |
| 18.1500 | -7.705e-01 |
| 18.2250 | -6.314e-01 |
| 18.3000 | -4.771e-01 |
| 18.3750 | -3.085e-01 |
| 18.4500 | -1.263e-01 |
| 18.5250 | 6.870e-02 |
| 18.6000 | 2.757e-01 |
| 18.6750 | 4.939e-01 |
| 18.7500 | 7.228e-01 |
| 18.8250 | 9.616e-01 |
| 18.9000 | 1.209e+00 |
| 18.9750 | 1.459e+00 |
| 19.0500 | 1.706e+00 |
| 19.1250 | 1.943e+00 |
| 19.2000 | 2.166e+00 |
| 19.2750 | 2.372e+00 |
| 19.3500 | 2.559e+00 |
| 19.4250 | 2.725e+00 |
| 19.5000 | 2.868e+00 |
| 19.7250 | 3.158e+00 |
| 19.8000 | 3.208e+00 |
| 19.8750 | 3.235e+00 |
| 19.9500 | 3.239e+00 |
| 20.0250 | 3.222e+00 |
| 20.1000 | 3.185e+00 |
| 20.1750 | 3.127e+00 |
| 20.2500 | 3.050e+00 |
| 20.3250 | 2.954e+00 |
| 20.4000 | 2.841e+00 |
| 20.4750 | 2.712e+00 |
| 20.5500 | 2.566e+00 |
| 20.6250 | 2.406e+00 |
| 20.7000 | 2.232e+00 |
| 20.7750 | 2.045e+00 |
| 20.8500 | 1.845e+00 |
| 20.9250 | 1.633e+00 |
| 21.0000 | 1.411e+00 |
| 21.0750 | 1.178e+00 |
| 21.1500 | 9.353e-01 |
| 21.2250 | 6.857e-01 |
| 21.3000 | 4.362e-01 |
| 21.3750 | 1.930e-01 |
| 21.4500 | -3.893e-02 |
| 21.5250 | -2.557e-01 |
| 21.6000 | -4.544e-01 |
| 21.6750 | -6.328e-01 |
| 21.7500 | -7.896e-01 |
| 21.8250 | -9.236e-01 |
| 21.9000 | -1.034e+00 |
| 21.9750 | -1.122e+00 |
| 22.0500 | -1.185e+00 |
| 22.1250 | -1.226e+00 |
| 22.2000 | -1.244e+00 |
| 22.2750 | -1.240e+00 |
| 22.3500 | -1.215e+00 |
| 22.4250 | -1.169e+00 |
| 22.5000 | -1.103e+00 |
| 22.5750 | -1.019e+00 |
| 22.6500 | -9.161e-01 |
| 22.7250 | -7.964e-01 |
| 22.8000 | -6.605e-01 |
| 22.8750 | -5.092e-01 |
| 22.9500 | -3.433e-01 |
| 23.0250 | -1.638e-01 |
| 23.1000 | 2.873e-02 |
| 23.1750 | 2.334e-01 |
| 23.2500 | 4.494e-01 |
| 23.3250 | 6.762e-01 |
| 23.4000 | 9.131e-01 |
| 23.4750 | 1.159e+00 |
| 23.5500 | 1.409e+00 |
| 23.6250 | 1.657e+00 |
| 23.7000 | 1.896e+00 |
| 23.7750 | 2.123e+00 |
| 23.8500 | 2.332e+00 |
| 23.9250 | 2.523e+00 |

| t[s] | Yk |
|---|---|
| 24.1500 | 2.966e+00 |
| 24.2250 | 3.067e+00 |
| 24.3000 | 3.145e+00 |
| 24.3750 | 3.199e+00 |
| 24.4500 | 3.231e+00 |
| 24.5250 | 3.240e+00 |
| 24.6000 | 3.228e+00 |
| 24.6750 | 3.194e+00 |
| 24.7500 | 3.140e+00 |
| 24.8250 | 3.067e+00 |
| 24.9000 | 2.975e+00 |
| 24.9750 | 2.865e+00 |
| 25.0500 | 2.739e+00 |
| 25.1250 | 2.597e+00 |
| 25.2000 | 2.440e+00 |
| 25.2750 | 2.268e+00 |
| 25.3500 | 2.083e+00 |
| 25.4250 | 1.886e+00 |
| 25.5000 | 1.677e+00 |
| 25.5750 | 1.456e+00 |
| 25.6500 | 1.225e+00 |
| 25.7250 | 9.846e-01 |
| 25.8000 | 7.359e-01 |
| 25.8750 | 4.858e-01 |
| 25.9500 | 2.409e-01 |
| 26.0250 | 6.349e-03 |
| 26.1000 | -2.137e-01 |
| 26.1750 | -4.162e-01 |
| 26.2500 | -5.988e-01 |
| 26.3250 | -7.600e-01 |
| 26.4000 | -8.986e-01 |
| 26.4750 | -1.014e+00 |
| 26.5500 | -1.106e+00 |
| 26.6250 | -1.174e+00 |
| 26.7000 | -1.220e+00 |
| 26.7750 | -1.242e+00 |
| 26.8500 | -1.243e+00 |
| 26.9250 | -1.222e+00 |
| 27.0000 | -1.180e+00 |
| 27.0750 | -1.118e+00 |
| 27.1500 | -1.037e+00 |
| 27.2250 | -9.380e-01 |
| 27.3000 | -8.217e-01 |
| 27.3750 | -6.889e-01 |
| 27.4500 | -5.406e-01 |
| 27.5250 | -3.776e-01 |
| 27.6000 | -2.007e-01 |
| 27.6750 | -1.076e-02 |
| 27.7500 | 1.915e-01 |
| 27.8250 | 4.053e-01 |
| 27.9000 | 6.300e-01 |
| 27.9750 | 8.649e-01 |
| 28.0500 | 1.109e+00 |
| 28.1250 | 1.359e+00 |
| 28.2000 | 1.608e+00 |
| 28.2750 | 1.849e+00 |
| 28.3500 | 2.079e+00 |
| 28.4250 | 2.292e+00 |
| 28.5000 | 2.487e+00 |
| 28.5750 | 2.661e+00 |
| 28.6500 | 2.813e+00 |
| 28.7250 | 2.943e+00 |
| 28.8000 | 3.049e+00 |
| 28.8750 | 3.131e+00 |
| 28.9500 | 3.190e+00 |
| 29.0250 | 3.226e+00 |
| 29.1000 | 3.240e+00 |
| 29.1750 | 3.232e+00 |
| 29.2500 | 3.202e+00 |
| 29.3250 | 3.152e+00 |
| 29.4000 | 3.083e+00 |
| 29.4750 | 2.995e+00 |
| 29.5500 | 2.889e+00 |
| 29.6250 | 2.766e+00 |
| 29.7000 | 2.626e+00 |
| 29.7750 | 2.472e+00 |
| 29.8500 | 2.303e+00 |
| 29.9250 | 2.121e+00 |
| 30.0000 | 1.926e+00 |
| 30.0750 | 1.719e+00 |
| 30.1500 | 1.501e+00 |
| 30.2250 | 1.272e+00 |
| 30.3000 | 1.033e+00 |
| 30.3750 | 7.860e-01 |
| 30.4500 | 5.356e-01 |
| 30.5250 | 2.892e-01 |
| 30.6000 | 5.221e-02 |
| 30.6750 | -1.710e-01 |
| 30.7500 | -3.772e-01 |
| 30.8250 | -5.640e-01 |
| 30.9000 | -7.295e-01 |
| 30.9750 | -8.727e-01 |
| 31.0500 | -9.928e-01 |
| 31.1250 | -1.089e+00 |
| 31.2000 | -1.163e+00 |
| 31.2750 | -1.213e+00 |
| 31.3500 | -1.240e+00 |
| 31.4250 | -1.244e+00 |
| 31.5000 | -1.227e+00 |
| 31.5750 | -1.190e+00 |
| 31.6500 | -1.132e+00 |
| 31.7250 | -1.055e+00 |
| 31.8000 | -9.592e-01 |
| 31.8750 | -8.463e-01 |
| 31.9500 | -7.167e-01 |
| 32.0250 | -5.715e-01 |
| 32.1000 | -4.113e-01 |
| 32.1750 | -2.372e-01 |
| 32.2500 | -4.976e-02 |
| 32.3250 | 1.501e-01 |
| 32.4000 | 3.617e-01 |
| 32.4750 | 5.843e-01 |
| 32.5500 | 8.172e-01 |
| 32.6250 | 1.060e+00 |
| 32.7000 | 1.309e+00 |
| 32.7750 | 1.559e+00 |
| 32.8500 | 1.802e+00 |
| 32.9250 | 2.034e+00 |
| 33.0000 | 2.251e+00 |
| 33.0750 | 2.449e+00 |
| 33.1500 | 2.628e+00 |
| 33.2250 | 2.785e+00 |
| 33.3000 | 2.919e+00 |
| 33.3750 | 3.029e+00 |
| 33.4500 | 3.117e+00 |
| 33.5250 | 3.180e+00 |
| 33.6000 | 3.221e+00 |
| 33.6750 | 3.239e+00 |
| 33.7500 | 3.235e+00 |
| 33.8250 | 3.210e+00 |
| 33.9000 | 3.164e+00 |
| 33.9750 | 3.098e+00 |
| 34.0500 | 3.014e+00 |
| 34.1250 | 2.911e+00 |
| 34.2000 | 2.791e+00 |
| 34.2750 | 2.655e+00 |
| 34.3500 | 2.504e+00 |
| 34.4250 | 2.338e+00 |
| 34.5000 | 2.159e+00 |
| 34.5750 | 1.966e+00 |
| 34.6500 | 1.762e+00 |
| 34.7250 | 1.546e+00 |
| 34.8000 | 1.319e+00 |
| 34.8750 | 1.082e+00 |
| 34.9500 | 8.359e-01 |
| 35.0250 | 5.855e-01 |
| 35.1000 | 3.378e-01 |
| 35.1750 | 9.862e-02 |
| 35.2500 | -1.276e-01 |
| 35.3250 | -3.375e-01 |
| 35.4000 | -5.283e-01 |
| 35.4750 | -6.982e-01 |
| 35.5500 | -8.459e-01 |
| 35.6250 | -9.707e-01 |
| 35.7000 | -1.072e+00 |
| 35.7750 | -1.150e+00 |
| 35.8500 | -1.204e+00 |
| 35.9250 | -1.236e+00 |
| 36.0000 | -1.245e+00 |
| 36.0750 | -1.232e+00 |
| 36.1500 | -1.199e+00 |
| 36.2250 | -1.145e+00 |
| 36.3000 | -1.072e+00 |
| 36.3750 | -9.797e-01 |
| 36.4500 | -8.702e-01 |
| 36.5250 | -7.439e-01 |
| 36.6000 | -6.017e-01 |
| 36.6750 | -4.445e-01 |
| 36.7500 | -2.731e-01 |
| 36.8250 | -8.827e-02 |
| 36.9000 | 1.092e-01 |
| 36.9750 | 3.185e-01 |
| 37.0500 | 5.389e-01 |
| 37.1250 | 7.698e-01 |
| 37.2000 | 1.010e+00 |
| 37.2750 | 1.259e+00 |
| 37.3500 | 1.509e+00 |

| t[s] | Yk |
|---|---|
| 37.4250 | 1.754e+00 |
| 37.5000 | 1.989e+00 |
| 37.5750 | 2.209e+00 |
| 37.6500 | 2.411e+00 |
| 37.7250 | 2.594e+00 |
| 37.8000 | 2.755e+00 |
| 37.8750 | 2.894e+00 |
| 37.9500 | 3.009e+00 |
| 38.0250 | 3.101e+00 |
| 38.1000 | 3.169e+00 |
| 38.1750 | 3.215e+00 |
| 38.2500 | 3.237e+00 |
| 38.3250 | 3.238e+00 |
| 38.4000 | 3.217e+00 |
| 38.4750 | 3.175e+00 |
| 38.5500 | 3.113e+00 |
| 38.6250 | 3.032e+00 |
| 38.7000 | 2.933e+00 |
| 38.7750 | 2.817e+00 |
| 38.8500 | 2.684e+00 |
| 38.9250 | 2.536e+00 |
| 39.0000 | 2.373e+00 |
| 39.0750 | 2.196e+00 |
| 39.1500 | 2.006e+00 |
| 39.2250 | 1.804e+00 |
| 39.3000 | 1.590e+00 |
| 39.3750 | 1.365e+00 |
| 39.4500 | 1.130e+00 |
| 39.5250 | 8.857e-01 |
| 39.6000 | 6.356e-01 |
| 39.6750 | 3.869e-01 |
| 39.7500 | 1.456e-01 |
| 39.8250 | -8.360e-02 |
| 39.9000 | -2.969e-01 |
| 39.9750 | -4.917e-01 |
| 40.0500 | -6.659e-01 |
| 40.1250 | -8.182e-01 |
| 40.2000 | -9.476e-01 |
| 40.2750 | -1.054e+00 |
| 40.3500 | -1.136e+00 |
| 40.4250 | -1.195e+00 |
| 40.5000 | -1.231e+00 |
| 40.5750 | -1.245e+00 |
| 40.6500 | -1.237e+00 |
| 40.7250 | -1.207e+00 |
| 40.8000 | -1.157e+00 |
| 40.8750 | -1.088e+00 |
| 40.9500 | -9.995e-01 |
| 41.0250 | -8.935e-01 |
| 41.1000 | -7.705e-01 |
| 41.1750 | -6.314e-01 |
| 41.2500 | -4.771e-01 |
| 41.3250 | -3.085e-01 |
| 41.4000 | -1.263e-01 |
| 41.4750 | 6.871e-02 |
| 41.5500 | 2.757e-01 |
| 41.6250 | 4.940e-01 |
| 41.7000 | 7.228e-01 |
| 41.7750 | 9.616e-01 |

| t[s] | Yk |
|---|---|
| 41.8500 | 1.209e+00 |
| 41.9250 | 1.459e+00 |
| 42.0000 | 1.706e+00 |
| 42.0750 | 1.943e+00 |
| 42.1500 | 2.166e+00 |
| 42.2250 | 2.372e+00 |
| 42.3000 | 2.559e+00 |
| 42.3750 | 2.725e+00 |
| 42.4500 | 2.868e+00 |
| 42.5250 | 2.988e+00 |
| 42.6000 | 3.084e+00 |
| 42.6750 | 3.158e+00 |
| 42.7500 | 3.208e+00 |
| 42.8250 | 3.235e+00 |
| 42.9000 | 3.239e+00 |
| 42.9750 | 3.222e+00 |
| 43.0500 | 3.185e+00 |
| 43.1250 | 3.127e+00 |
| 43.2000 | 3.050e+00 |
| 43.2750 | 2.954e+00 |
| 43.3500 | 2.841e+00 |
| 43.4250 | 2.712e+00 |
| 43.5000 | 2.566e+00 |
| 43.5750 | 2.406e+00 |
| 43.6500 | 2.232e+00 |
| 43.7250 | 2.045e+00 |
| 43.8000 | 1.845e+00 |
| 43.8750 | 1.633e+00 |
| 43.9500 | 1.411e+00 |
| 44.0250 | 1.178e+00 |
| 44.1000 | 9.353e-01 |
| 44.1750 | 6.857e-01 |
| 44.2500 | 4.362e-01 |
| 44.3250 | 1.930e-01 |
| 44.4000 | -3.893e-02 |
| 44.4750 | -2.557e-01 |
| 44.5500 | -4.544e-01 |
| 44.6250 | -6.328e-01 |
| 44.7000 | -7.896e-01 |
| 44.7750 | -9.236e-01 |
| 44.8500 | -1.034e+00 |
| 44.9250 | -1.122e+00 |
| 45.0000 | -1.185e+00 |
| 45.0750 | -1.226e+00 |
| 45.1500 | -1.244e+00 |
| 45.2250 | -1.240e+00 |
| 45.3000 | -1.215e+00 |
| 45.3750 | -1.169e+00 |
| 45.4500 | -1.103e+00 |
| 45.5250 | -1.019e+00 |
| 45.6000 | -9.161e-01 |
| 45.6750 | -7.964e-01 |
| 45.7500 | -6.605e-01 |
| 45.8250 | -5.092e-01 |
| 45.9000 | -3.433e-01 |
| 45.9750 | -1.638e-01 |
| 46.0500 | 2.873e-02 |
| 46.1250 | 2.334e-01 |
| 46.2000 | 4.494e-01 |

| t[s] | Yk |
|---|---|
| 46.2750 | 6.762e-01 |
| 46.3500 | 9.131e-01 |
| 46.4250 | 1.159e+00 |
| 46.5000 | 1.409e+00 |
| 46.5750 | 1.657e+00 |
| 46.6500 | 1.896e+00 |
| 46.7250 | 2.123e+00 |
| 46.8000 | 2.332e+00 |
| 46.8750 | 2.523e+00 |
| 46.9500 | 2.693e+00 |
| 47.0250 | 2.841e+00 |
| 47.1000 | 2.966e+00 |
| 47.1750 | 3.067e+00 |
| 47.2500 | 3.145e+00 |
| 47.3250 | 3.199e+00 |
| 47.4000 | 3.231e+00 |
| 47.4750 | 3.240e+00 |
| 47.5500 | 3.228e+00 |
| 47.6250 | 3.194e+00 |
| 47.7000 | 3.140e+00 |
| 47.7750 | 3.067e+00 |
| 47.8500 | 2.975e+00 |
| 47.9250 | 2.865e+00 |
| 48.0000 | 2.739e+00 |
| 48.0750 | 2.597e+00 |
| 48.1500 | 2.440e+00 |
| 48.2250 | 2.268e+00 |
| 48.3000 | 2.083e+00 |
| 48.3750 | 1.886e+00 |
| 48.4500 | 1.677e+00 |
| 48.5250 | 1.456e+00 |
| 48.6000 | 1.225e+00 |
| 48.6750 | 9.846e-01 |
| 48.7500 | 7.359e-01 |
| 48.8250 | 4.858e-01 |
| 48.9000 | 2.409e-01 |
| 48.9750 | 6.350e-03 |
| 49.0500 | -2.137e-01 |
| 49.1250 | -4.162e-01 |
| 49.2000 | -5.988e-01 |
| 49.2750 | -7.600e-01 |
| 49.3500 | -8.986e-01 |
| 49.4250 | -1.014e+00 |
| 49.5000 | -1.106e+00 |
| 49.5750 | -1.174e+00 |
| 49.6500 | -1.220e+00 |
| 49.7250 | -1.242e+00 |
| 49.8000 | -1.243e+00 |
| 49.8750 | -1.222e+00 |
| 49.9500 | -1.180e+00 |
| 50.0250 | -1.118e+00 |
| 50.1000 | -1.037e+00 |
| 50.1750 | -9.380e-01 |
| 50.2500 | -8.217e-01 |
| 50.3250 | -6.889e-01 |
| 50.4000 | -5.406e-01 |
| 50.4750 | -3.770e-01 |
| 50.5500 | -2.007e-01 |
| 50.6250 | -1.076e-02 |

| t[s] | Yk |
|---|---|
| 50.7000 | 1.915e-01 |
| 50.7750 | 4.053e-01 |
| 50.8500 | 6.300e-01 |
| 50.9250 | 8.649e-01 |
| 51.0000 | 1.109e+00 |
| 51.0750 | 1.359e+00 |
| 51.1500 | 1.608e+00 |
| 51.2250 | 1.849e+00 |
| 51.3000 | 2.079e+00 |
| 51.3750 | 2.292e+00 |
| 51.4500 | 2.487e+00 |
| 51.5250 | 2.661e+00 |
| 51.6000 | 2.813e+00 |
| 51.6750 | 2.943e+00 |
| 51.7500 | 3.049e+00 |
| 51.8250 | 3.131e+00 |
| 51.9000 | 3.190e+00 |
| 51.9750 | 3.226e+00 |
| 52.0500 | 3.240e+00 |
| 52.1250 | 3.232e+00 |
| 52.2000 | 3.202e+00 |
| 52.2750 | 3.152e+00 |
| 52.3500 | 3.083e+00 |
| 52.4250 | 2.995e+00 |
| 52.5000 | 2.889e+00 |
| 52.5750 | 2.766e+00 |
| 52.6500 | 2.626e+00 |
| 52.7250 | 2.472e+00 |
| 52.8000 | 2.303e+00 |
| 52.8750 | 2.121e+00 |
| 52.9500 | 1.926e+00 |
| 53.0250 | 1.719e+00 |
| 53.1000 | 1.501e+00 |
| 53.1750 | 1.272e+00 |
| 53.2500 | 1.033e+00 |
| 53.3250 | 7.860e-01 |
| 53.4000 | 5.356e-01 |
| 53.4750 | 2.892e-01 |
| 53.5500 | 5.221e-02 |
| 53.6250 | -1.710e-01 |
| 53.7000 | -3.772e-01 |
| 53.7750 | -5.640e-01 |
| 53.8500 | -7.295e-01 |
| 53.9250 | -8.727e-01 |
| 54.0000 | -9.928e-01 |
| 54.0750 | -1.089e+00 |
| 54.1500 | -1.163e+00 |
| 54.2250 | -1.213e+00 |
| 54.3000 | -1.240e+00 |
| 54.3750 | -1.244e+00 |
| 54.4500 | -1.227e+00 |
| 54.5250 | -1.190e+00 |
| 54.6000 | -1.132e+00 |
| 54.6750 | -1.055e+00 |
| 54.7500 | -9.592e-01 |
| 54.8250 | -8.463e-01 |
| 54.9000 | -7.167e-01 |
| 54.9750 | -5.715e-01 |
| 55.0500 | -4.113e-01 |

| t[s] | Yk |
|---|---|
| 55.1250 | -2.372e-01 |
| 55.2000 | -4.976e-02 |
| 55.2750 | 1.501e-01 |
| 55.3500 | 3.617e-01 |
| 55.4250 | 5.843e-01 |
| 55.5000 | 8.172e-01 |
| 55.5750 | 1.060e+00 |
| 55.6500 | 1.309e+00 |
| 55.7250 | 1.559e+00 |
| 55.8000 | 1.802e+00 |
| 55.8750 | 2.034e+00 |
| 55.9500 | 2.251e+00 |
| 56.0250 | 2.449e+00 |
| 56.1000 | 2.628e+00 |
| 56.1750 | 2.785e+00 |
| 56.2500 | 2.919e+00 |
| 56.3250 | 3.029e+00 |
| 56.4000 | 3.117e+00 |
| 56.4750 | 3.180e+00 |
| 56.5500 | 3.221e+00 |
| 56.6250 | 3.239e+00 |
| 56.7000 | 3.235e+00 |
| 56.7750 | 3.210e+00 |
| 56.8500 | 3.164e+00 |
| 56.9250 | 3.098e+00 |
| 57.0000 | 3.014e+00 |
| 57.0750 | 2.911e+00 |
| 57.1500 | 2.791e+00 |
| 57.2250 | 2.655e+00 |
| 57.3000 | 2.504e+00 |
| 57.3750 | 2.338e+00 |
| 57.4500 | 2.159e+00 |
| 57.5250 | 1.966e+00 |
| 57.6000 | 1.762e+00 |
| 57.6750 | 1.546e+00 |
| 57.7500 | 1.319e+00 |
| 57.8250 | 1.082e+00 |
| 57.9000 | 8.359e-01 |
| 57.9750 | 5.855e-01 |
| 58.0500 | 3.378e-01 |
| 58.1250 | 9.862e-02 |
| 58.2000 | -1.276e-01 |
| 58.2750 | -3.375e-01 |
| 58.3500 | -5.283e-01 |
| 58.4250 | -6.982e-01 |
| 58.5000 | -8.459e-01 |
| 58.5750 | -9.707e-01 |
| 58.6500 | -1.072e+00 |
| 58.7250 | -1.150e+00 |
| 58.8000 | -1.204e+00 |
| 58.8750 | -1.236e+00 |
| 58.9500 | -1.245e+00 |
| 59.0250 | -1.232e+00 |
| 59.1000 | -1.199e+00 |
| 59.1750 | -1.145e+00 |
| 59.2500 | -1.072e+00 |
| 59.3250 | -9.797e-01 |
| 59.4000 | -8.702e-01 |
| 59.4750 | -7.439e-01 |

| t[s] | Yk |
|---|---|
| 59.5500 | -6.017e-01 |
| 59.6250 | -4.445e-01 |
| 59.7000 | -2.731e-01 |
| 59.7750 | -8.827e-02 |
| 59.8500 | 1.092e-01 |
| 59.9250 | 3.185e-01 |

*Ejercicio # 3*

Si se emplea la misma planta del ejercicio #2, pero se cambia la alinealidad del relé ideal por una saturación ideal de valores U=0.2 y D=1, se presenta el siguiente resultado en el gráfico de G(jw) con -1/N.

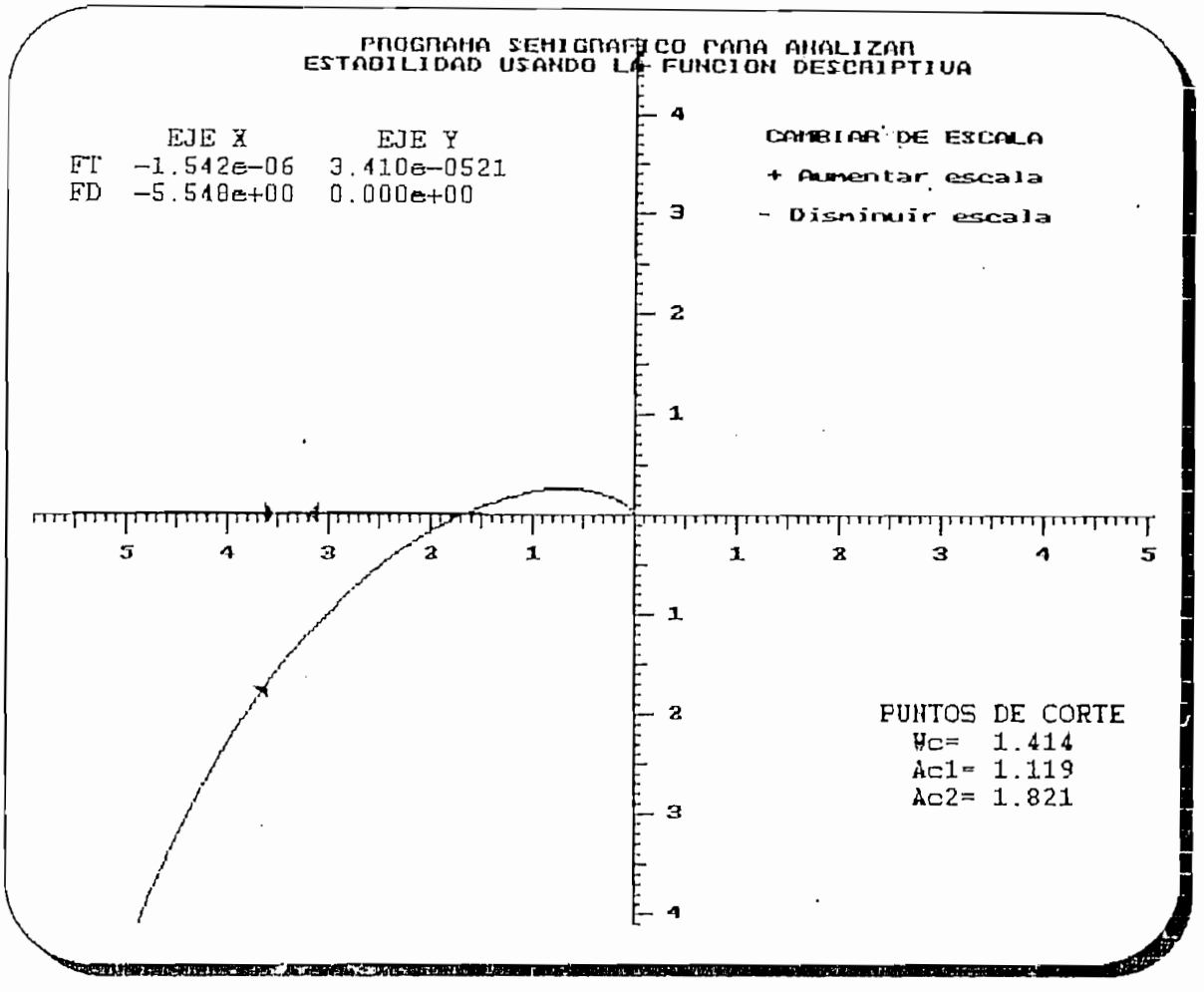



*Fig. 3.6 Diagrama de -1/N y G(jw)*

Se aprecia que existen dos puntos de cruce, esto provoca que aparezcan dos ciclos límites, el primero inestable (Wc=1.414 Ac1=1.119), y el segundo estable (Wc=1.414 Ac2=1.821).

Como puede verse, la curva -1/N, empieza desplazandose desde la izquierda hacia la derecha en el eje real negativo, pero llega a un punto

de coordenadas (-1.5,0) en donde invierte el sentido de desplazamiento, dirigiéndose entonces desde la derecha hacia la izquierda, este desplazamiento en el programa es muy rápido por lo que no se lo puede apreciar fácilmente, es por ello que es muy útil el listado de valores de la respuesta de -1/N, con los cuales se podrá ver que la curva -1/N pasa dos veces por el cruce en el cual G(jw) tiene parte imaginaria igual a cero.

Puesto que existen dos ciclos límites, resulta muy importante el escoger correctamente las condiciones iniciales del sistema, ya que de ello depende que llegue o no a la estabilidad, tal es así que si las condiciones iniciales colocan al punto inicial en una zona de ciclo límite estable, la salida se mantendrá oscilando como en los ejercicios anteriores, pero si el punto inicial está dentro de un ciclo límite inestable, la salida o tiende a estabilizarse o a desplazarse hacia el ciclo límite estable, esto se ve en la figura 3.6 en donde si nos encontramos en el ciclo límite inestable, pero con un punto inicial a la derecha del punto de cruce, estaremos en una región inestable por lo cual la amplitud tiende a crecer y nos desplazaría el punto hacia el ciclo límite estable, al contrario si el punto inicial se encuentra a la izquierda del punto de cruce inestable, nos encontraremos en una región estable ya que la curva G(jw) no encierra al punto inicial, tendiendo entonces la amplitud a disminuir llegando así a la estabilidad; esto se aprecia en la figura 3.7 en donde se ha puesto como condiciones iniciales cero, llegando con ello a la estabilidad del sistema.

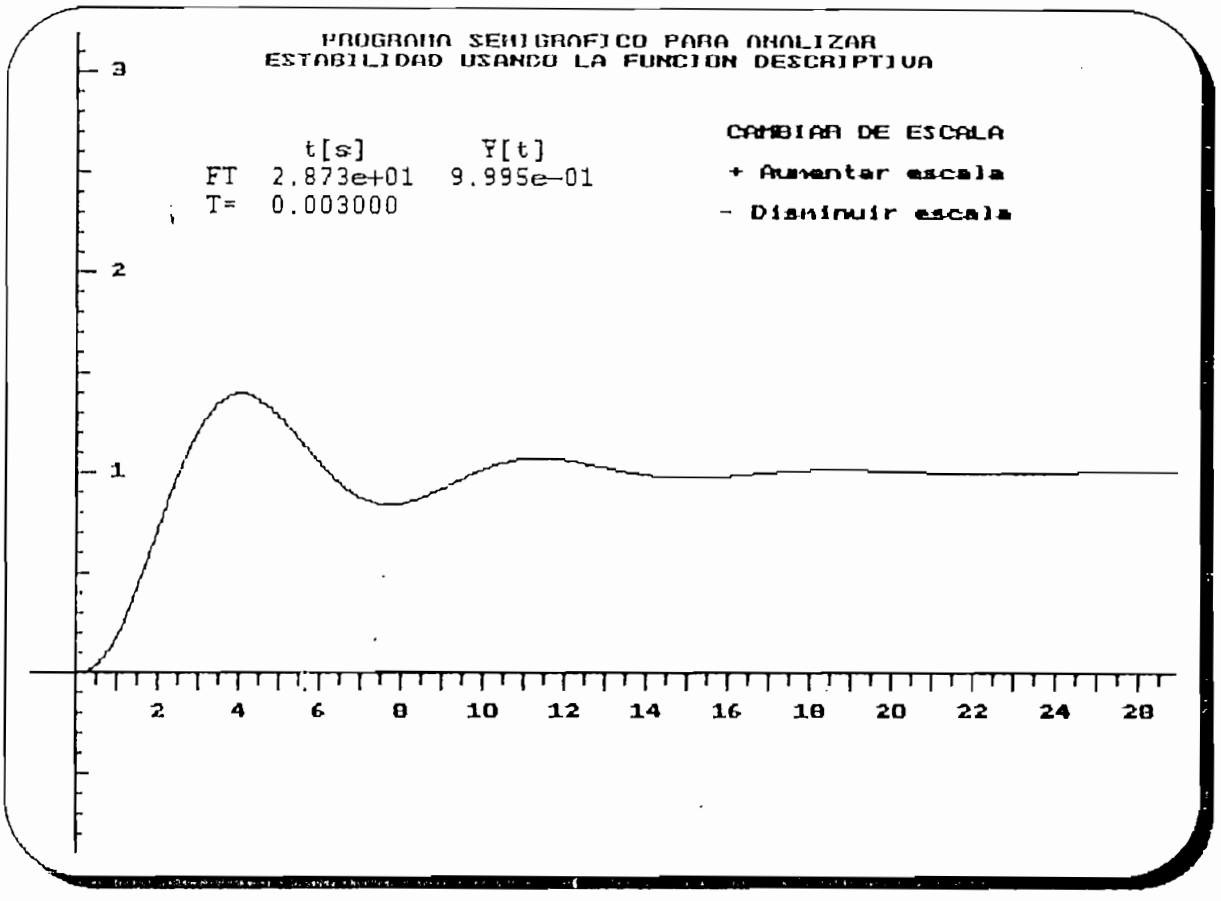



*Fig. 3.7 Respuesta en el tiempo para el ciclo límite estable*

Se puede ver que en el presente caso la salida no permanece oscilando, y se estabiliza en el valor de uno, debido a que estamos en el ciclo límite inestable pero con condiciones iniciales que llevan al sistema a la estabilidad, entonces mediante la introducción de esta alinealidad se convierte el sistema que originalmente era inestable en uno estable. Se puede ver que el sistema era inestable, ya que la gráfica de G(jw) de la planta encierra al punto -1+j0, lo cual indica que este era un sistema inestable.

## 3.2 EJERCICIOS PARA PROBAR LIMITACIONES DEL PROGRAMA

Las limitaciones que presentaría este programa, son por ejemplo cuando se tenga un sistema demasiado rápido, en donde la determinación del período de muestreo es de suma importancia, ya que si el período de muestreo es demasiado grande para las características del sistema, se perdería información, y la respuesta en el tiempo que presenta el programa no estaría apegada a la realidad.

Pero si se escoge un período de muestreo muy pequeño, de manera que se obtenga una información real del sistema, se presenta el grave inconveniente de que el tiempo máximo del cual se puede obtener el gráfico es proporcional al período de muestreo de la siguiente manera:

$$tmax = 20000*T$$

En donde T es el período de muestreo, por lo tanto si el período es muy pequeño, ejemplo T=0.0001, el tiempo máximo que graficará el programa es de tmax=2 segundos, lo cual puede ser insuficiente para apreciar si se ha estabilizado o no un sistema.

El hecho de tener esta limitación es que se pone un tope máximo de 20000 puntos para muestreo de la señal de salida, pero si bien es cierto se podría ampliar este valor, con lo cual se graficarían más períodos de tiempo en la pantalla, pero ello trae consigo que el programa se vuelva más lento, razón por la cual resulta necesario llegar a un compromiso de cuanto sería este número de muestras que se empleen.

A continuación se demuestra esta limitación por medio de un ejemplo de un sistema muy rápido, mostrado en la siguiente figura.

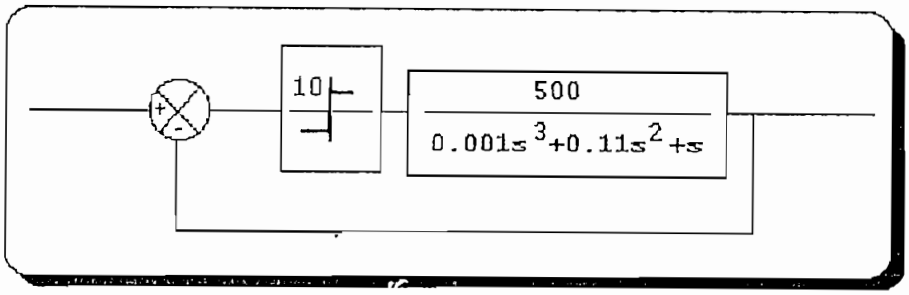



*Fig. 3.8 Sistema con No-Linealidad ON-OFF* [1]

Esta planta resulta ser muy rápida como se aprecia en la figura 3.9 que corresponde a su respuesta en el tiempo.

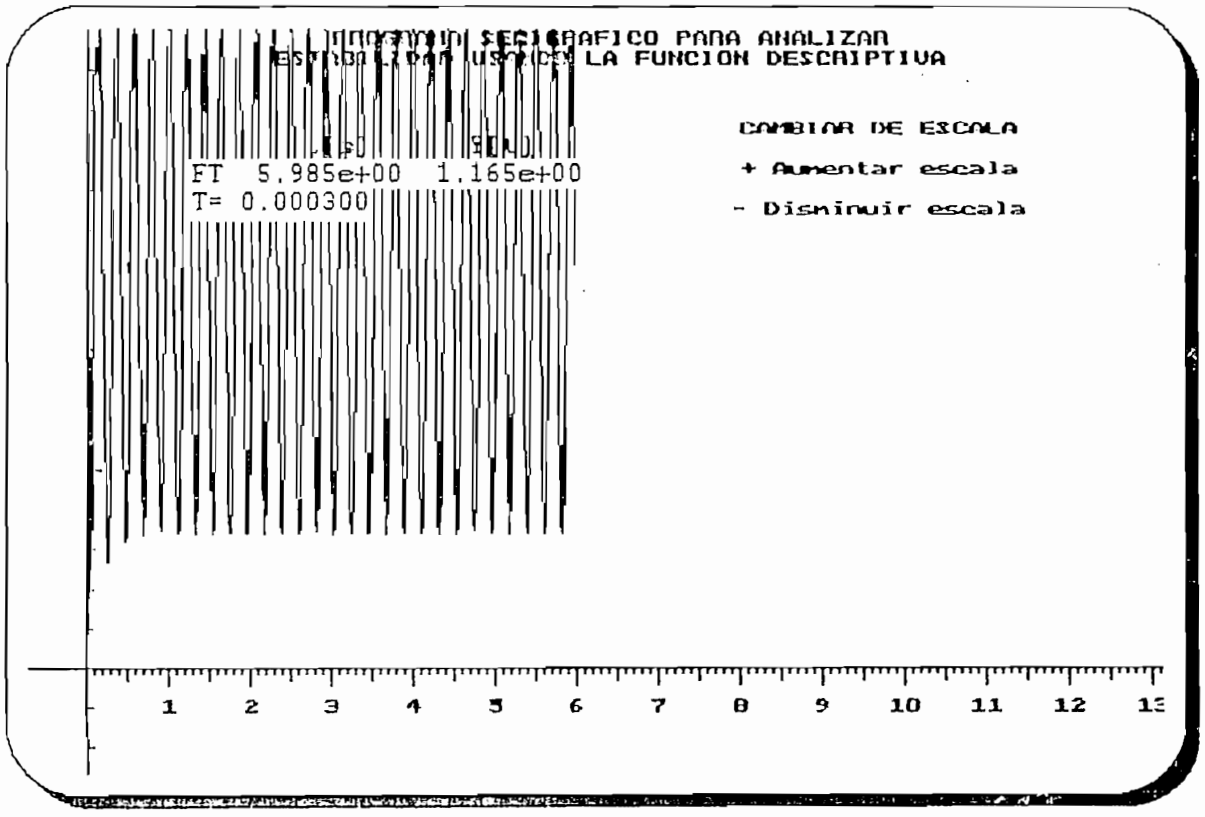



*Fig. 3.9 Respuesta en el tiempo del sistema de la figura 3.8*

---

1 ECHEVERRIA Pablo, Análisis de Sistemas de Control No-Lineales utilizando la Función Descriptiva, EPN, 1990, pags 170-172

El período empleado en este caso es igual a 0.0003, entonces como ya se indicó antes sólo se podrá hacer un muestreo hasta el tiempo igual a 6 segundos que correspondería a los 20000 puntos que se toma de muestra.

Para poder obtener la respuesta de este sistema hasta un tiempo mayor a 6 segundos es necesario aumentar el período de muestreo, pero se corre el riesgo de que se obtenga una señal distorsionada debido a la rapidez de este sistema.

Otra limitación que presenta este programa es que la compensación que se puede emplear debe ser del tipo PID, por lo tanto no se podría emplear redes de adelanto o atraso, pero se puede solucionar este problema colocando el compensador después de la alinealidad en lugar de ser antes de ella, con lo cual se consigue tener después de la alinealidad dos funciones de transferencia de las cuales se obtendría una equivalente, poniendo en el programa en el momento de ingresar la función de transferencia de la planta, la función del sistema equivalente planta-compensador, esto aunque sale del esquema planteado en esta tesis produce resultados aceptables.

Un ejemplo de esto lo constituye el siguiente sistema.

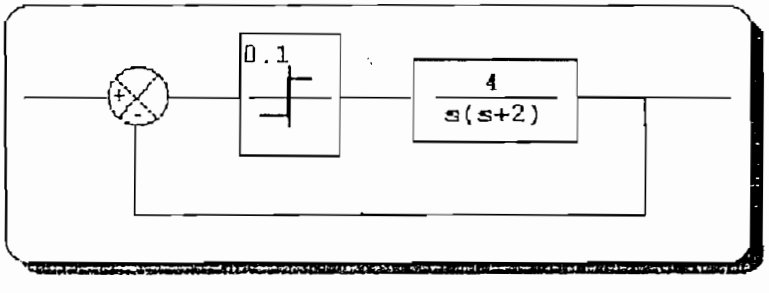



*Fig. 3.10 Sistema estable con alinealidad ON-OFF*

La respuesta en el tiempo para este caso se muestra en la figura 3.11, en donde se observa que el sistema tiende a estabilizarse, aunque presenta una pequeña oscilación, debido a que el ciclo límites que se presenta en este caso es inestable, pero la amplitud tiende a cero, mientras que la frecuencia tiende a infinito.

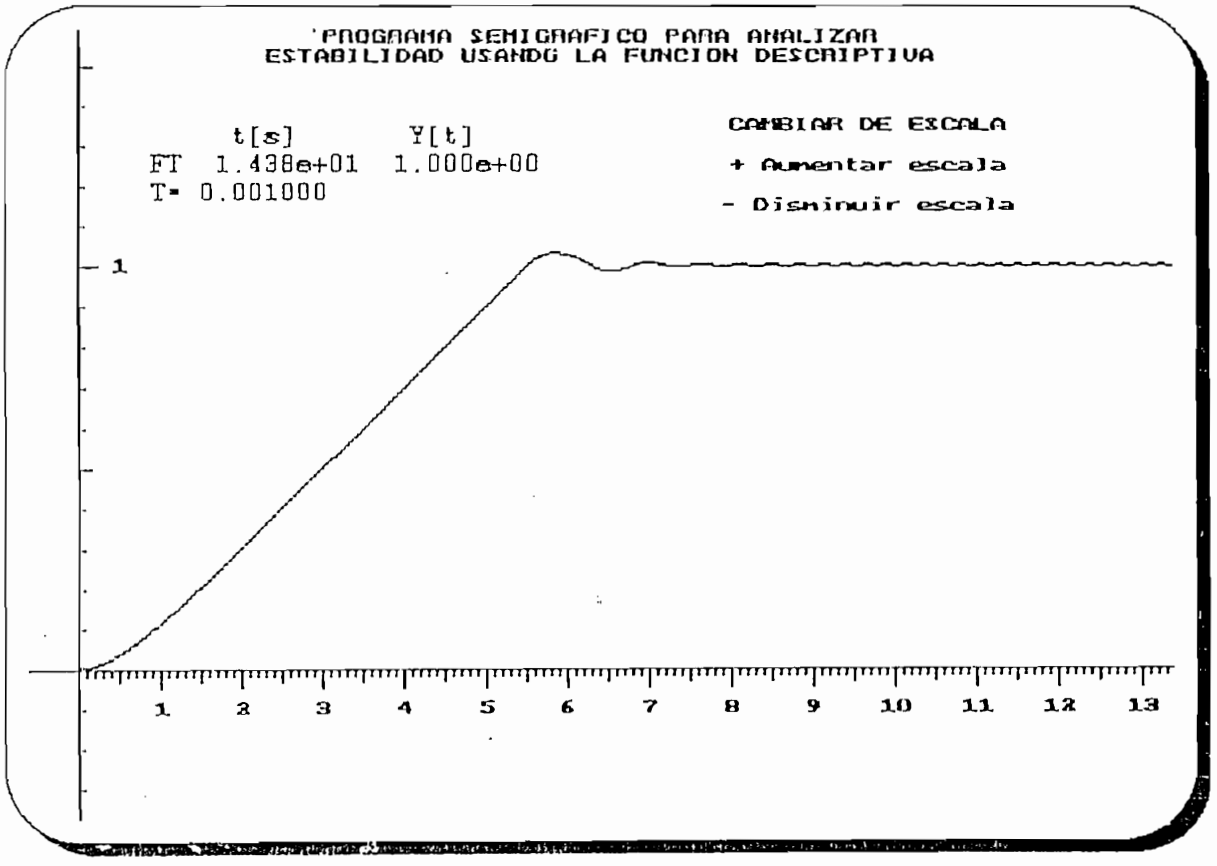



*Fig. 3.11 Respuesta en el tiempo del sistema de la figura 3.10*

Pero si al sistema de la figura 3.10 se le desea colocar una red de adelanto con la finalidad de que el tiempo de estabilidad sea menor, se lo hará poniendo la compensación después de la alinealidad.

Si se emplea una compensación igual a:

$$Gc = \frac{4.675\times(s+2.9)}{(s+5.4)}$$

Entonces

$$G(s) * Gc(s) = \frac{18.7\times(s+2.9)}{s(s+2)(s+5.4)}$$

Mediante la implementación de esta nueva planta compensada en el programa se obtiene una respuesta en el tiempo, que se estabiliza más rápidamente como se aprecia en la figura 3.12

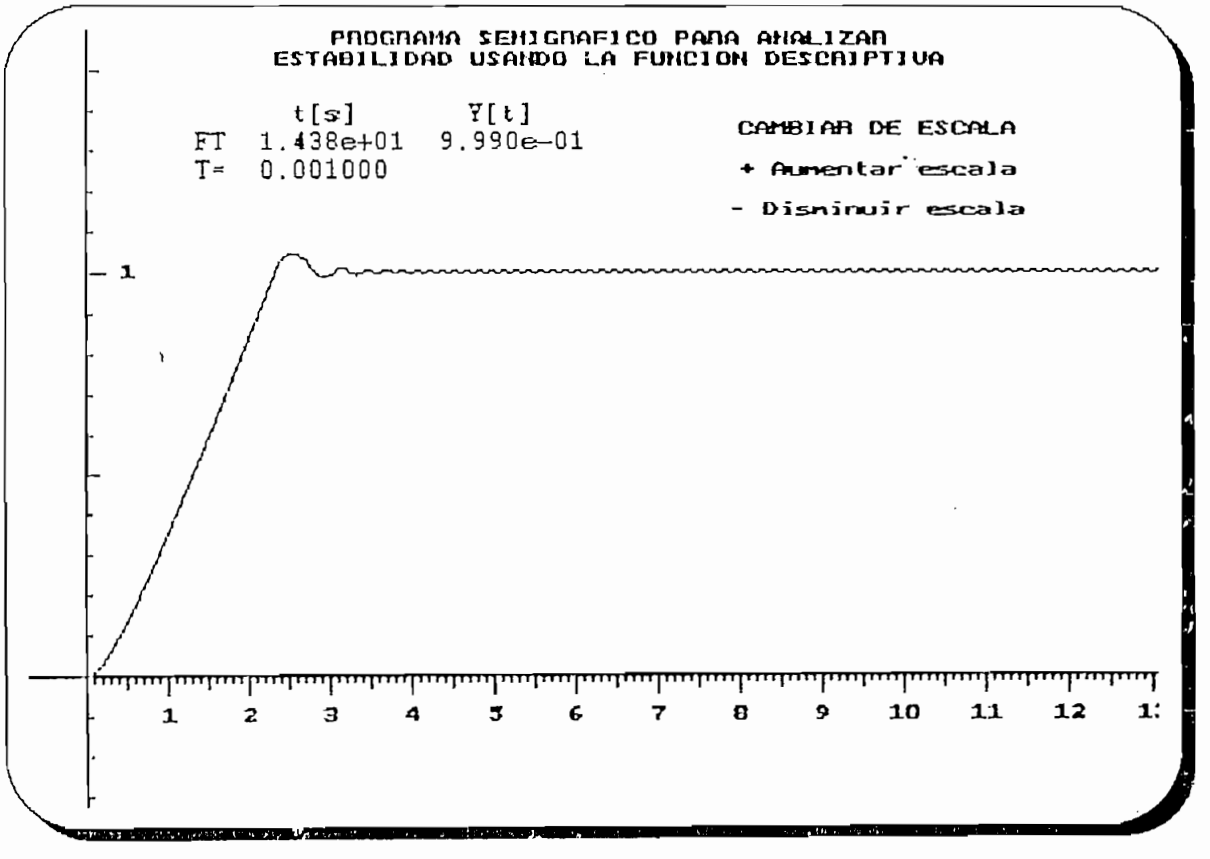



*Fig. 3.12 Respuesta en el tiempo del sistema compensado*

## 3.3 EJERCICIOS PARA PROBAR LA CALIDAD DEL PROGRAMA

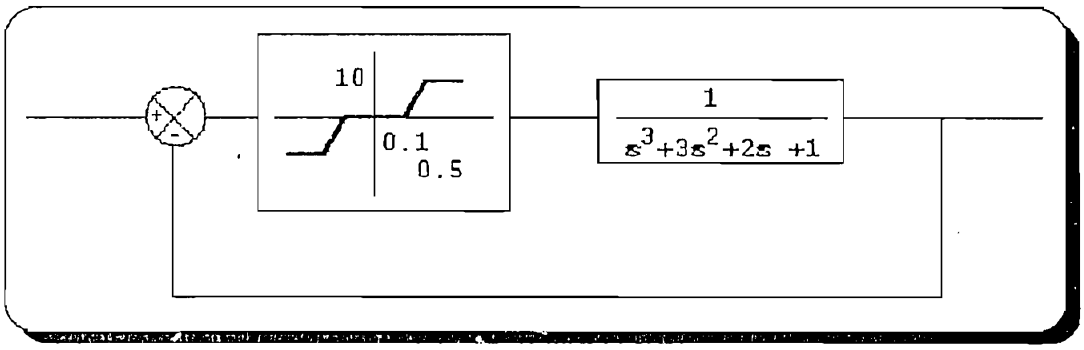



*Fig. 3.13 Planta con Saturación con Zona Muerta*

Aquí se presenta un sistema compuesto por una planta de tercer orden y una alinealidad del tipo saturación con zona muerta, sistema que presenta un ciclo límite estable de valores Wc=1.414 y Ac=2.526. Se trata pues de un ciclo estable por el sentido que tiene el desplazamiento de las dos curvas mostradas en la figura 3.14, por lo tanto en este caso si se realiza una graficación en el tiempo aparecerá una salida oscilante de frecuencia 1.414 y amplitud 2.526.

Para corregir este problema de inestabilidad en el sistema resulta conveniente eliminar el ciclo límite, esto se lo logra entonces moviendo o cambiando la característica de la curva G(jw) mostrada en la figura 3.14, aquí pues es de suma utilidad el poder emplear un PID como compensación, con lo cual se podría evitar el ciclo límite.

En este caso se ha colocado un compensador tipo PD ya que con el se lograría acelerar el sistema eliminando entonces el ciclo límite.

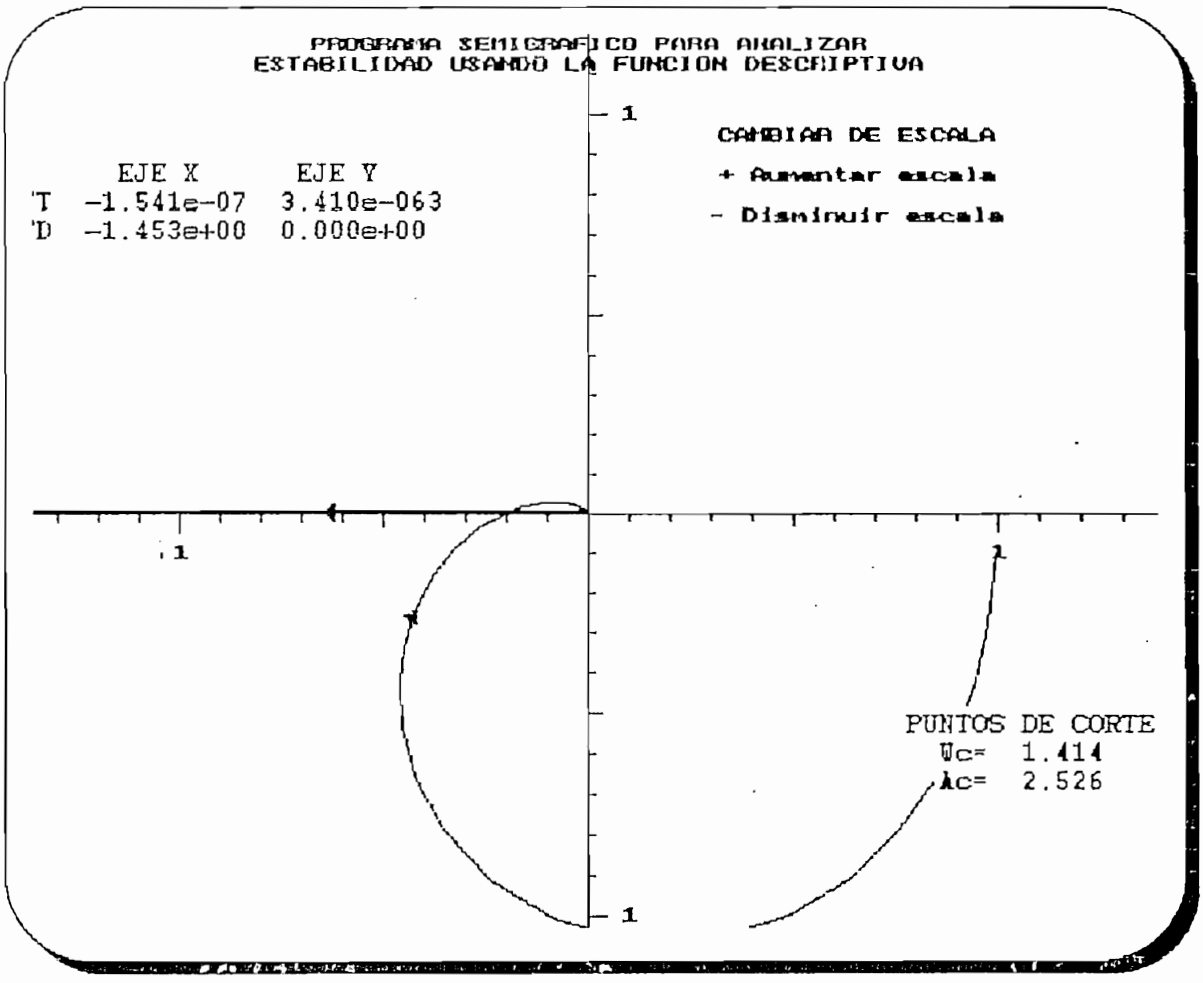



*Fig. 3.14 Diagrama de G(jw) y -1/N del sistema de la figura 3.13*

El compensador empleado es:

$$Gc(s) = 0.8 + 0.3 \times s$$

Con la implementación de este compensador antes de la alinealidad se consigue obtener un sistema estable ya que no hay punto de cruce entre las dos curva G(jw) y -1/N, pero el sistema presenta una respuesta transitoria subamortiguada ya que muestra un sobre pulso, pero se estabiliza rápidamente, figura 3.15.

Es necesario destacar que el tiempo de estabilización como el máximo sobreimpulso se puede también mejorar empleando otros valores para el compensador PID.

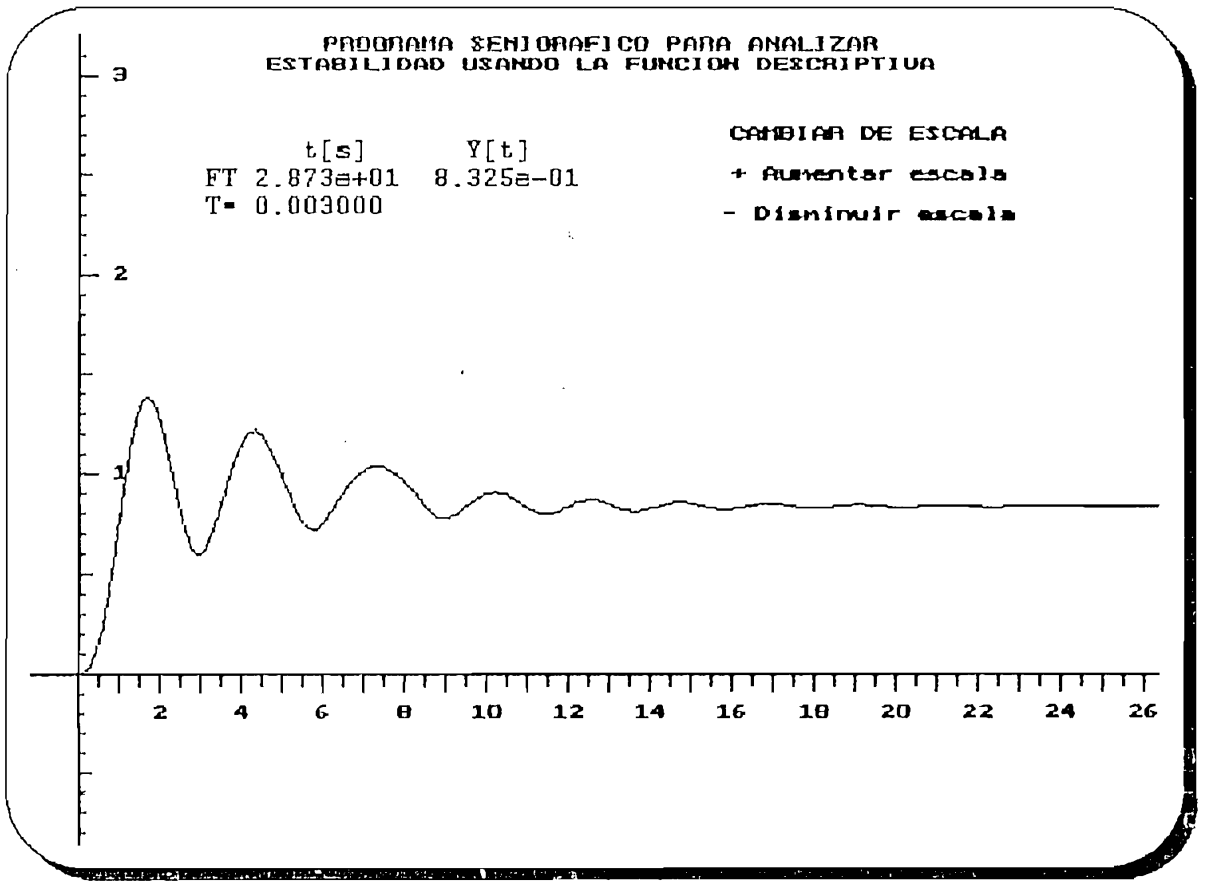



*Fig. 3.15 Respuesta en el tiempo del sistema de la fig. 3.13 compensado*

# CAPITULO IV

## CONCLUSIONES

# CAPITULO IV

## 4.1 CONCLUSIONES

De acuerdo a la teoría desarrollada que sirve de base para este trabajo, y varios ejercicios realizados para probar los resultados que se obtiene con este programa, de los cuales solo una pequeña parte se muestran en esta tesis, se ha llegado a establecer las siguientes conclusiones:

- La selección del tipo de control que debe utilizarse para cierta planta y los elementos que se requieren introducir en ella (alinealidades ~~intensionales~~), así como también la determinación de las señales de entrada al sistema se basa principalmente en el grado de conocimiento que se tenga del mismo.

- En la presente simulación no se considera la posibilidad de que se introduzca ruido significante en el sistema una vez que éste ya está funcionando.

- Mediante este programa de simulación se ha conseguido evitar el tedioso trabajo de realizar todo un proceso de operaciones matemáticas repetitivas, lográndose entonces utilizar el tiempo en

analizar los resultados que se obtienen con el programa en lugar de estar calculándolos varias veces.

- En el método de discretización empleado para la función de transferencia de la planta, es muy importante el escoger correctamente el período de discretización, ya que de el depende el obtener una curva en el tiempo lo más cercana a la realidad.

- Si se realiza una ajuste de ganancia en el sistema utilizando para ello un compensador proporcional, se puede lograr que G(jw) ya no rodee a la curva -1/N, pero si esto no fuese posible por alguna razón indeseable se puede diseñar redes de atraso-adelanto o adelanto-atraso, como también el usar un controlador PID, para cambiar la forma de la curva G(jw) de tal manera que no rodee a la curva -1/N, y el sistema se vuelva estable.

- Para estos sistemas no lineales se pueden emplear las mismas redes de compensación que las usadas en los sistemas lineales, pero con la diferencia que el método de diseño de las mismas es completamente distinto, siendo preferible usar métodos de análisis en frecuencia cuando se usan sistemas no-lineales.

- Siempre es necesario realizar el diseño de un sistema no lineal sobre una base gráfico (Diagrama de Nyquist) puesto que la frecuencia y amplitud de las oscilaciones varían según se manipulen las gráficas de G(jw) y -1/N.

- Este método de la función descriptiva resulta muy útil para predecir la estabilidad de los sistemas no lineales, pero no facilita información alguna en lo que se refiere a su respuesta transitoria, es por ello que se ha incluido en el programa la graficación en el tiempo de la respuesta del sistema, para obtener entonces una información gráfica de la respuesta transitoria del mismo.

- En el análisis de la función descriptiva, la naturaleza de la alinealidad presente en el sistema, determina la complejidad del análisis.

- La exactitud que se puede obtener del análisis por medio del presente programa es mejor cuando se emplean sistemas de orden elevado, ya que para dichos sistemas se cumple de mejora manera la suposición inicial de que las plantas generalmente son filtros pasabajos.

Analizando gráficamente cada tipo de alinealidad se observa las siguientes características importantes:

- Se puede presentar casos en los cuales la planta inicialmente sea estable, es decir la curva G(jw) no encierre al punto -1+j0 en el diagrama de Nyquist, pero al aumentarle una alinealidad el sistema puede volverse inestable, ya que en este caso la estabilidad no se define con ver si la curva G(jw) encierra o no al punto -1+j0, sino

más bien en analizar si encierra a la curva -1/N o una porción de la misma.

- En los casos que se tiene alinealidades con zona muerta o histéresis, es de suma importancia que las condiciones iniciales del sistema proporcionen una entrada a la alinealidad mayor que la zona muerta o a la histéresis ya que de lo contrario la salida de la alinealidad se mantendrá en cero.

- Todos los tipos de relés y saturaciones analizados en esta tesis, presentan una curva de -1/N con parte imaginaria igual a cero, razón por la cual su representación gráfica es una recta sobre el eje real negativo en el plano complejo.

- En el caso de una histéresis ideal también su representación gráfica de -1/N es una recta pero desplazada del eje real un cierto valor que depende tanto de la amplitud de la histéresis como del valor de salida de la alinealidad (U)

- Cuando se tiene una alinealidad tipo Relé Ideal la curva -1/N se desplaza desde el punto (0,0) hacia $(-\infty,0)$, pero según crece el valor de U se generan generalmente ciclos límites estables de amplitud cada vez más grande, pero el tiempo que le toma al sistema en llegar al ciclo límite estable es cada vez menor.

- En el caso de tener un Relé con Zona Muerta se pueden obtener tanto un ciclo límite estable como uno inestable, esto es por que la curva -1/N se desplaza desde el punto $(-\infty,0)$ acercándose al eje jw, pero llega una momento en el cual cambia de sentido de giro yendo nuevamente al punto $(-\infty,0)$, este punto mínimo en el cual cambia de sentido la curva depende del valor de U/D, ya que si este crece el punto de inflexión de la curva se acerca más el eje jw, y viceversa.

- Si se usa un relé con Pendiente la curva de -1/N se desplaza desde el eje jw hacia valores negativos del eje real, pero el punto final de esta recta se acerca al eje jw conforme se aumente el valor de la pendiente K

- Estando empleando un Relé con Zona Muerta y Pendiente, la curva se mueve de $(-\infty,0)$ acercándose al eje jw, y se acerca más conforme la pendiente K aumente.

- En el caso de una Saturación Ideal o con Zona Muerta el comportamiento es similar ya que según la pendiente K aumente la curva -1/N se acerca al eje jw, y el sistema tiende más rápidamente hacia el ciclo límite que en estos casos es generalmente estable, solamente que la pendiente K es distinta para cada uno de estos casos siendo igual a U/D para la Saturación Ideal e igual a $U/(D_2-D_1)$ en el caso de una Saturación con Zona Muerta.

- En la saturación con pendiente hay dos pendientes que influyen en la respuesta la una U/D que se comporta igual a lo dicho en los casos anteriores, pero la pendiente $K_2$ (fig. 1.19), influye en forma distinta, ya que si esta pendiente crece el punto final de la curva -1/N se acerca al punto inicial, pudiendo cambiar el sentido de desplazamiento de la curva si este valor de $K_2$ es sumamente grande, con lo cual se puede convertir un sistema con ciclo límite estable a uno con ciclo límite inestable.

  Cosa similar sucede con la Saturación con Zona Muerta y Pendiente.

- Si la planta tiene una Histéresis Ideal, la amplitud del ciclo límite aumenta conforme la histéresis D también lo hace, pero su frecuencia disminuye.

- En los demás casos de Histéresis el comportamiento es muy similar, ya que la amplitud del ciclo límite aumenta conforme la pendiente también lo hace, pero la frecuencia del ciclo límite crece igualmente, ya que debe tomarse en cuenta que estos últimos tres casos de histéresis presentan curvas de -1/N y no rectas.

- El programa motivo de esta tesis fue desarrollado principalmente en lenguaje "C", ya que solo una pequeña parte como es la introducción musical y la salida del programa que es del mismo tipo fueron desarrolladas en Quick Basic, esto se lo realizó debido a las facilidades que presenta dicho lenguaje para este tipo de programas.

- Se eligió el lenguaje "C" para esta tesis por la facilidad que presenta en cuanto a portabilidad, ya que con solo tener el manejador de pantalla necesario para cada computadora, se puede correrlo fácilmente en cualquier compatible con IBM, ya que el programa automáticamente determina el tipo de tarjeta de gráficos que se disponga, trabajando entonces sin ningún problema.

- Finalmente con esta tesis se ha cumplido las metas y objetivos planteados como son, el poder contar con un programa que servirá de sustento para el estudio teórico que se brinda en la facultad dentro del pénsum de las materias de Control.

# BIBLIOGRAFIA

# BIBLIOGRAFIA


- CUNNINGHAM W. J., Introduction to Nonlinear Analisis, MsGraw-Hill, 1958
- COSGRIFF, Robert, Nonlinear Control Sistems, McGraw-Hill, 1958
- GIBSON John, Nonlinear Automatic Control, McGraw-Hill, 1963
- ELDGERD Olle, Control Systems Theory, McGraw-Hill, 1967
- KWAKERNAAK Huibert, Linear Optimal Control Systems, 1972
- OGATA Katsuiko, Ingeniería de Control Moderna, Prentice Hall, 1985
- DORF Richard, Sistemas Automáticos de Control, Fondo Educativo Latinoamericano, 1978
- KUO B, Automatic Control Systems, Prentice Hall, 1967

- ECHEVERRIA Pablo, Análisis de Sistemas de Control No-Lineales utilizando la Función Descriptiva, EPN, 1990

- PARRA Angel, Programa Digital para el Análisis de Sistemas No Lineales Mediante El Trazo de Trayectorias de Fase hasta en tres dimensiones, EPN, 1992

- PINTO Germánico, Identificación de Sistemas en Tiempo Real, EPN, 1991

- BORLAND, Turbo Vision for C++, 1991
- BORLAND, Turbo C++ Programer's Guide, 1990
- KERNIGHAM Brian, El lenguaje de programación C, Prentice-Hall, 1992
- BANDA Hugo, PROGRAMMING IN C, UNIVERSITY OF DUNDEE 1991

# ANEXOS

# ANEXO A

## MANUAL DE UTILIZACION DEL PROGRAMA

# ANEXO A

## MANUAL DE UTILIZACION DEL PROGRAMA

En este punto se trata de dar una guía de como utilizar de mejor manera el programa motivo de la presente tesis.

El programa básico se encuentra formado por tres archivos ejecutables llamados EFD.EXE, M_FINAL.EXE y M_INICIO.EXE, de los cuales los dos últimos son programas tipo musical que sirven de adorno al programa principal EFD.EXE (Estabilidad Función Descriptiva); además están presentes algunos archivos con extensión .BGI, los mismos que permiten al programa adaptarse a cualquier tipo de tarjeta de gráficos que se disponga en el computador en que se ejecuta el programa, estos archivos son:

ATT.BGI, CGA.BGI, EGAVGA.BGI, HERC.BGI, IBM8514.BGI, PC3270.BGI

Por lo tanto es de suma importancia que en el directorio donde esté colocado el programa principal EFD.EXE se encuentre también el manejador gráfico de pantalla adecuado.

El programa puede ser ejecutado colocando directamente su nombre desde el sistema operativo, por ejemplo:

C:\CONTROL\EFD.EXE

Inmediatamente de esto aparece una pantalla de presentación figura A.1, la misma que se la desactiva presionando un ENTER o mediante el uso del MOUSE presionando la tecla OK que aparece en dicho mensaje, quedando entonces el programa listo para ser utilizado.

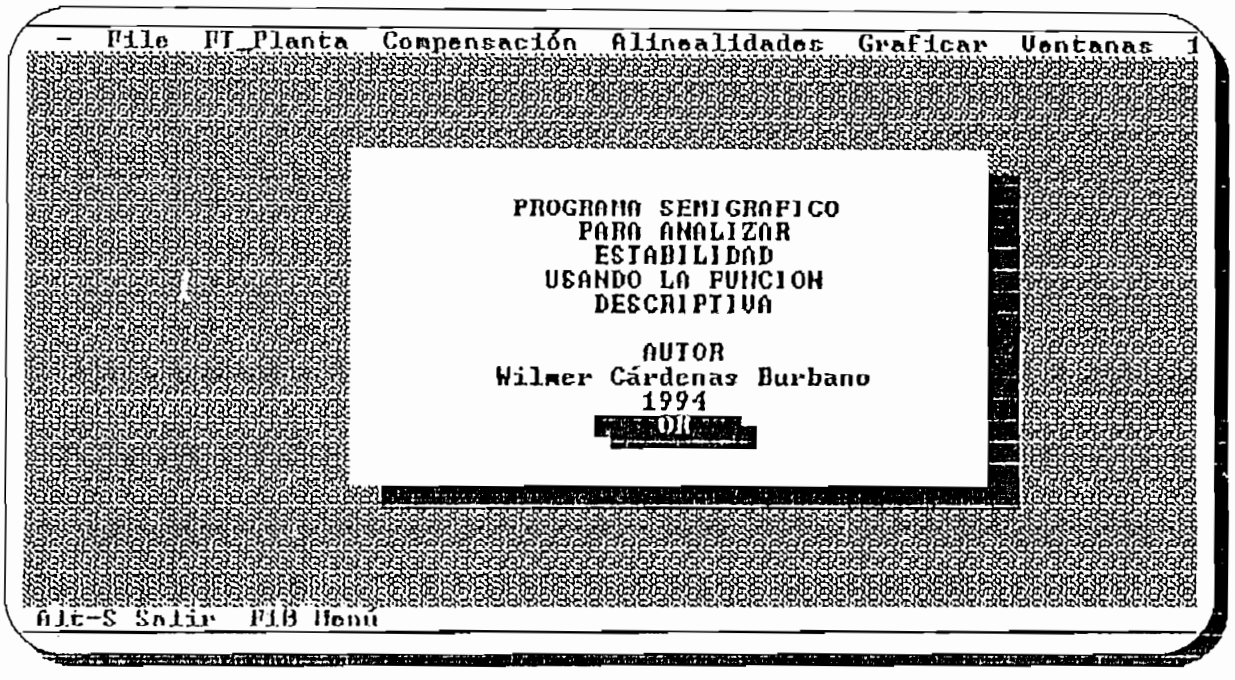



*Fig. A1 Pantalla de presentación .*

Luego de desactivar la pantalla de presentación, es necesario indicar que se tienen dos barras de menú una en la parte superior y otra en la parte inferior de la pantalla; en la parte inferior se indica que si se *presiona* las teclas Alt-S se sale del programa, mientras que con la tecla de función F10 se puede acceder al menú superior, pero a este menú también se puede acceder directamente con el MOUSE presionando el item que se desee analizar.

El área superior de la pantalla como se aprecia en la figura A1, tiene 7 menús principales; -, File, FT_Planta, Compensación, Alinealidad, Graficar, Ventanas

A continuación se hará un análisis de cada uno de estos menús indicando las características de cada uno.

(-)

Este símbolo indica que existe un submenú en este punto, pero que es de uso especial, y no afecta al desarrollo del programa, al escoger este menú aparece un submenú con dos opciones, la primera *Título*, que muestra la pantalla de presentación, figura A.1; y la segunda opción de este submenú *Calculadora*, despliega una pequeña calculadora que puede ser de mucha utilidad para realizar pequeños cálculos.

(File)

Mediante esta opción se tiene un submenú muy importante mostrado en la figura A.2, que como se ve existen cuatro opciones dentro de este menú que son:

*Abrir.-* Presenta en pantalla archivos de tipo ASCII que se encuentren grabados en el disco, esto es útil para analizar por ejemplo el comportamiento de los puntos del gráfico de -1/N.

*Grabar.-* En este caso se muestra otro submenú con cuatro opciones que son:

*Resp. Freq. FT,* Graba la respuesta de la Función de transferencia de la planta en una archivo

*Resp. Feq. Gc* Graba la respuesta de la Función de transferencia de la planta compensada en una archivo

*Resp. Freq. FD.* Graba la tabla de valores de -1/N ante variaciones de la amplitud A

*Resp. Tiempo* Graba la tabla de valores del gráfico de la respuesta en el tiempo

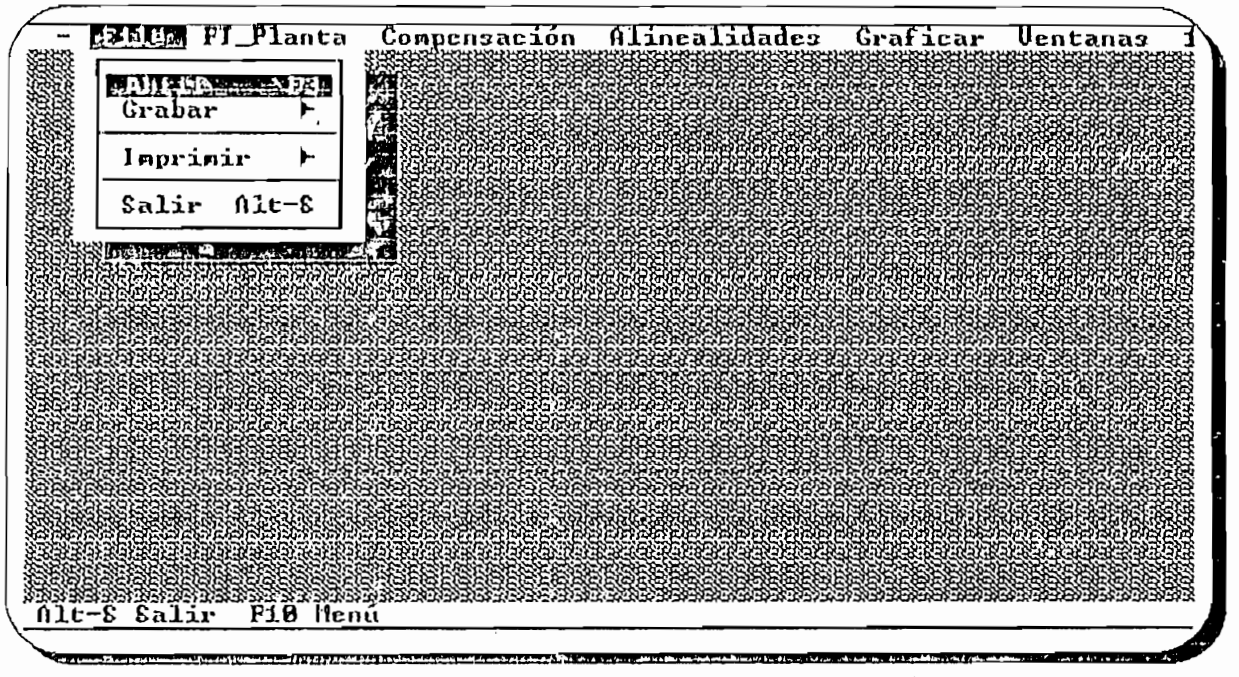



*Fig. A.2 Menú File*

*Imprimir.-* Es muy similar al comando grabar, pero en lugar de enviar los datos a un archivo lo hace llevándolos a la impresora, para lo cual esta debe esta encendida y puesta en línea con el computador, si esto no

sucede, el programa presentará un mensaje de que la impresora no se encuentra lista, o existe algún problema en la comunicación entre la computadora y la impresora.

Se debe indicar que las opciones de Grabar e Imprimir inicialmente se encuentran deshabilitadas, es decir que no se las puede usar, pero son automáticamente habilitadas cuando se han realizado los gráficos que involucran a la opción que se desea grabar o imprimir, así por ejemplo si se desea grabar la tabla de valores de la respuesta de un cierto sistema en el tiempo, se debe primeramente graficar la respuesta en el tiempo de dicho sistema.

*Salir.-* Por último aparece esta opción que como ya se indicó permite abandonar el programa.

(FT_Planta)

Si se escoge en el menú principal FT_Planta, se podrá ingresar la función de transferencia de la planta, la misma que puede ingresarse de dos maneras, una colocando los datos manualmente, u otra indicando que se utilice una función de transferencia grabada anteriormente, para ello se despliega un submenú con estas posibilidades.

*FT Nueva.-* El programa cambia de tipo de pantalla, y permite el ingreso de los coeficientes de la función de transferencia que se desee, un ejemplo de esto es la figura A.3.

Como puede verse se realiza una verificación de los datos ingresados, para tener entonces la posibilidad de corregirlos en caso de que se hayan ingresado mal los coeficientes.

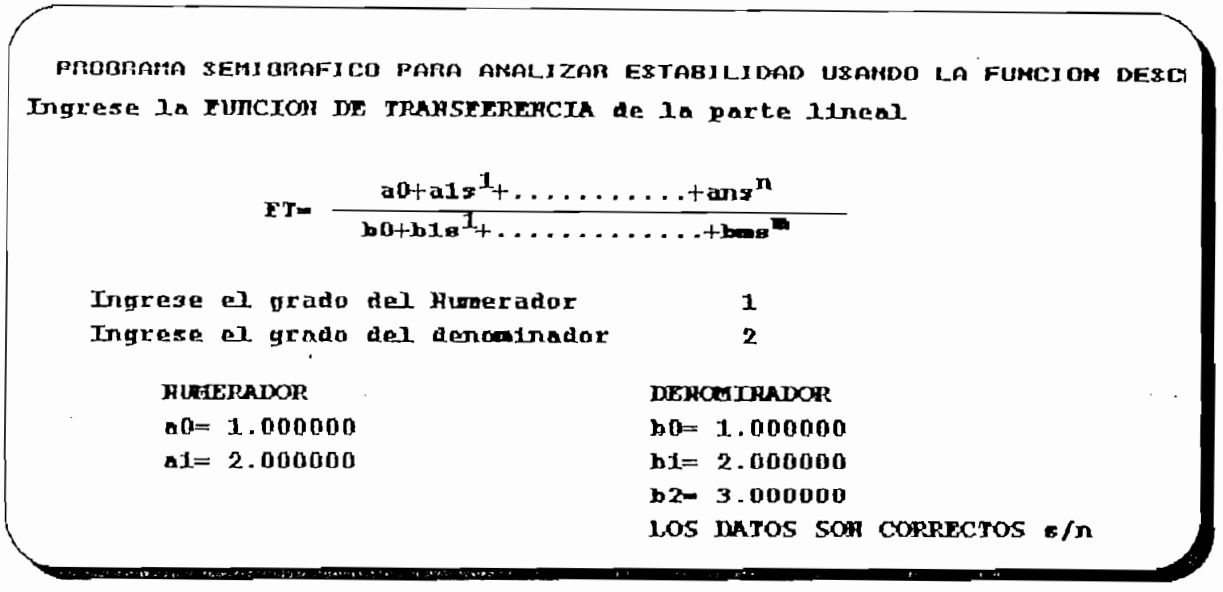



*Fig. A3 Ingreso de datos de la FT de la planta*

*FT Grabada.-* El programa pide en este caso que se de el nombre del archivo con el cual fue grabada la función de transferencia, pero el nombre del archivo debe ser puesto sin extensión, ya que ella está predefinida en el programa, esta pantalla para ingresar el nombre de la función de transferencia a utilizarse se muestra en la figura A4.

El momento de ingresar el nombre del archivo a usarse se puede también ingresar el path (dirección) de donde se encuentre el dicho archivo, por ejemplo A:\control\programas\ejer1

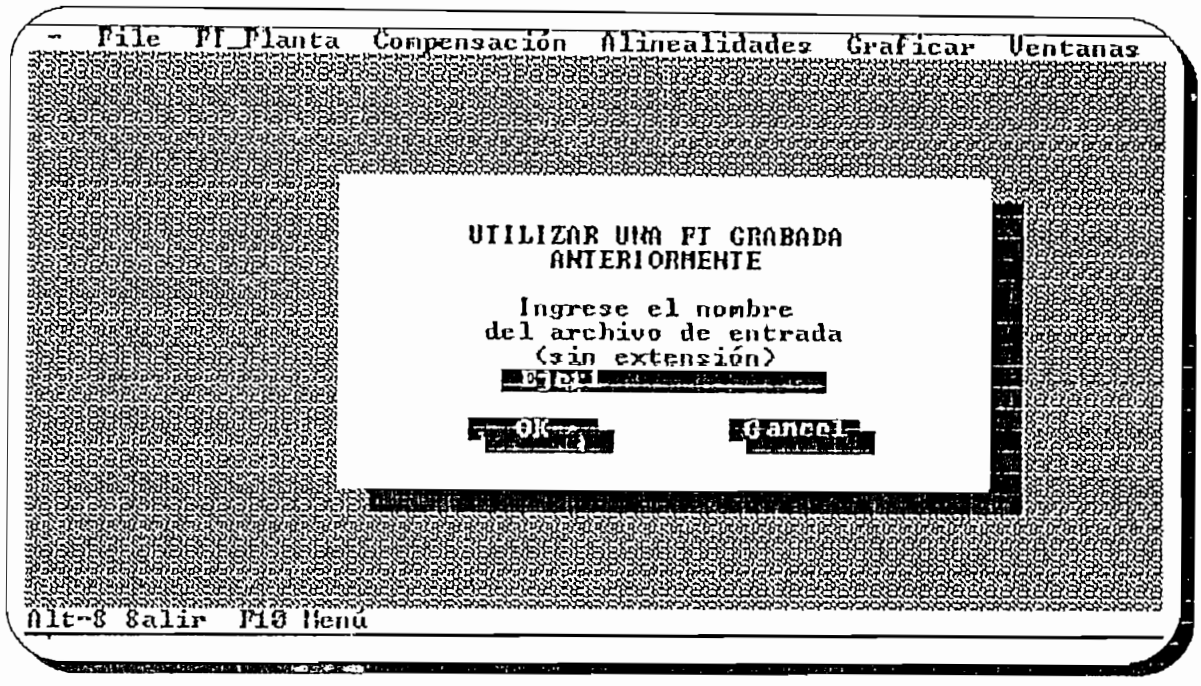



*Fig. A.4 Ingreso de una FT grabada anteriormente*

*Ver la FT.*- Esta es La última opción que se presenta en el submenú FT_Planta la cual permite observar en cualquier momento cual es la función de transferencia que se está utilizando, y su presentación es igual a la figura A3.

(Compensación)

Tal como lo indica su nombre, permite el ingreso de una compensación tipo PID, pero solo podrá ser ingresada si antes se ha colocado una función de transferencia de la planta, entonces igual como en el ingreso de la FT de la planta se lo puede hacer de dos formas, ingresando los coeficientes del PID manualmente *Gc Nueva*, o usando una compensación tipo PID grabada en el disco *Gc Grabada*, también aparece al final la opción *Ver la Gc*, para mostrar en pantalla la compensación que se

está empleando; todas estas tres opciones son muy similares a las empleadas con la FT de la planta.

*Eliminar la Gc.-* En este caso existe esta opción que sirve para que el programa considere que no se ha ingresado compensación alguna; cuando se escoge este comando, solo se presenta un pequeño parpadeo en la pantalla, ya que no se realiza ninguna confirmación, por lo que hay que tener mucho cuidado con el uso de esta opción.

(Alinealidades)

Aquí se puede escoger una alinealidad dentro de un grupo que se tiene para ello, entonces se desplegará un submenú con cinco opciones figura A.5

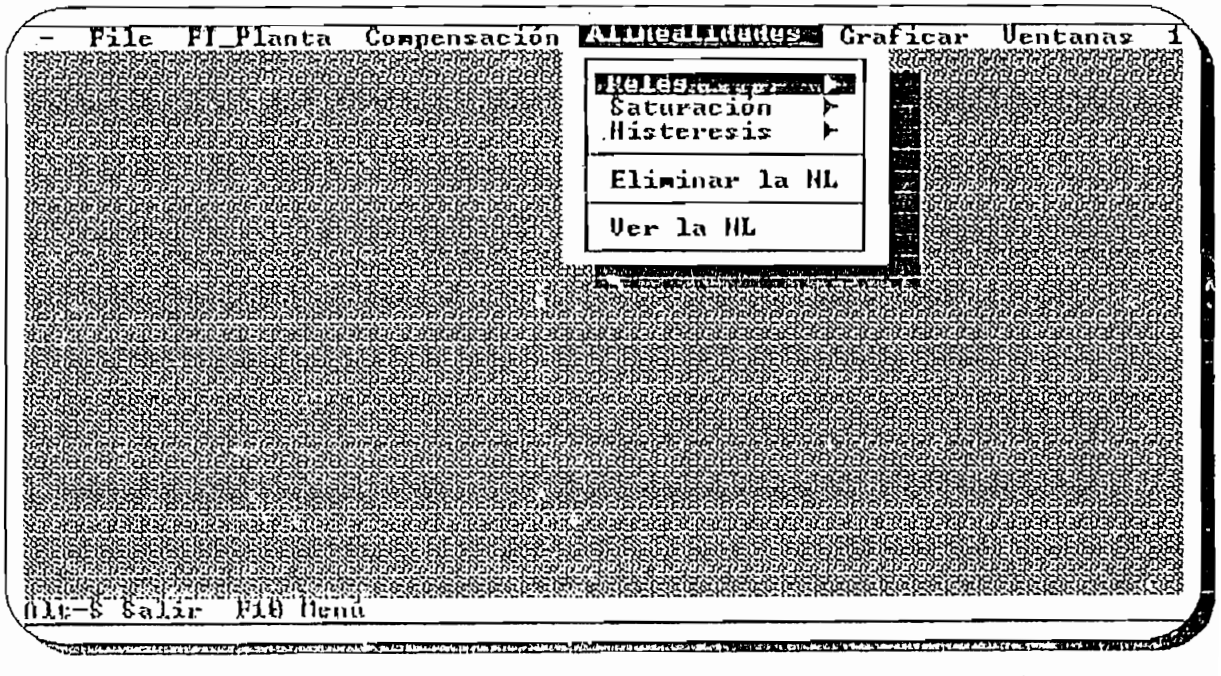



*Fig. A5 Ventana para escoger el tipo de Alinealidad*

Las tres primeras opciones que presenta este menú son para escoger el tipo de alinealidad, que puede estar dentro de este grupo, como es *Relés, Saturación, Histéresis*, pero al escoger cual es el caso de alinealidad a utilizar se despliega un submenú que tiene a su ves cuatro opciones *Ideal, con Zona Muerta, con Pendiente, con ZM. y Pend.*, estas cuatro opciones se presentan para cada una de las tres principales Relés, Saturación, Histéresis, teniéndose por lo tanto un total de 12 tipos de alinealidades posibles a escoger.

Una vez escogido cual es el tipo de alinealidad a usarse, el programa pedirá que se ingresen los datos de dicha alinealidad para lo cual muestra un dibujo en la pantalla de la forma de la alinealidad, y pide los datos de ella.

Igual como en el caso de ingresar una compensación también aquí se tiene la posibilidad de *Eliminar la NL* de la memoria del computador.

También se muestra la opción *Ver la NL.*, que permite revisar en cualquier momento, que tipo de alinealidad se tiene escogida

(GRAFICAR)

Mediante este comando se obtiene una pantalla que muestra cuatro formas de graficación, que son:

*Graf. -1/N y G(jw).-* Permite obtener un gráfico de estas dos curvas, y presenta el punto de cruce de las mismas si es que existe, pero solo se puede acceder a esta opción de graficación si se ha escogido una función de transferencia de la planta como también una alinealidad.

*Graf. -1/N y Gc(jw).-* En este caso se obtiene un gráfico similar al anterior, pero en lugar de graficar G(jw), se está graficando G(jw)*PID, dando como resultado una función de transferencia compensada; resulta útil el tener estas dos opciones independientes una de la otra para poder hacer el análisis de los dos gráficos, a esta opción de igual manera que a la anterior solo se podrá acceder si se tiene ingresada una compensación y una alinealidad.

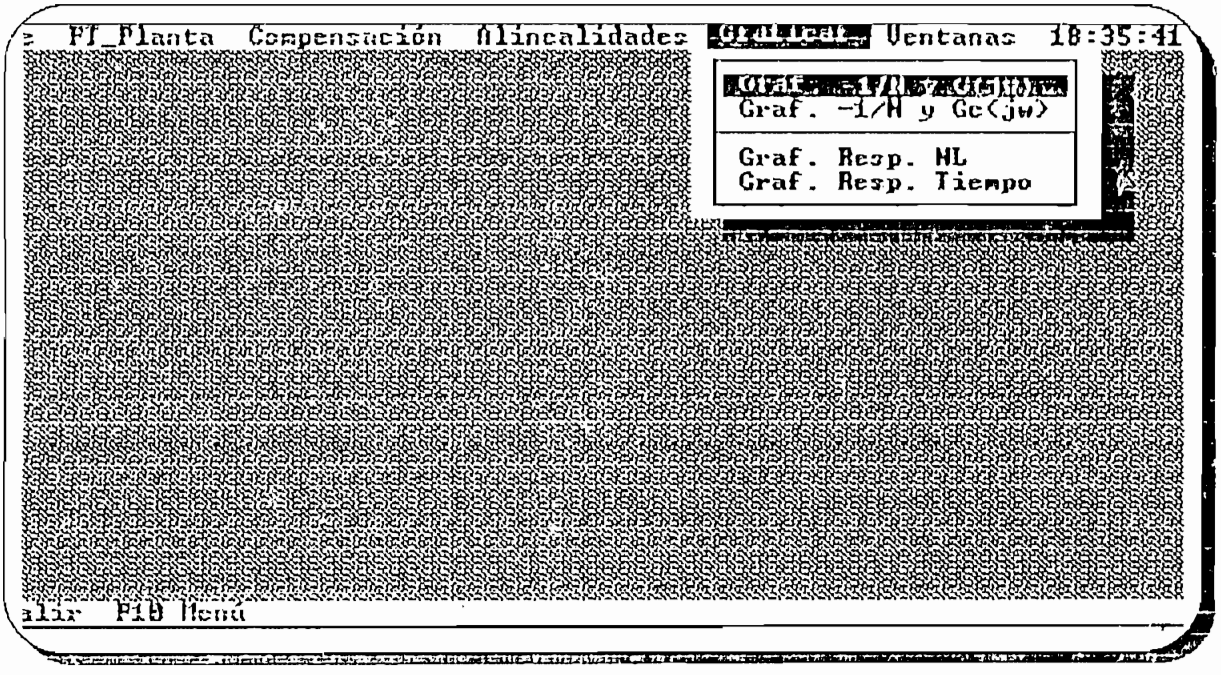



*Fig. A6 Formas de graficar*

*Graf. Resp. NL.-* Mediante este caso se permite el graficar la respuesta que presentaría solamente la alinealidad escogida ante una entrada sinusoidal de amplitud y frecuencia constantes.

*Graf. Resp. Tiempo.-* Permite obtener la respuesta del sistema en el tiempo, pero para ello es necesario ingresar primeramente el período de muestreo, como también las condiciones iniciales que estarían contenidas en el vector de estado.

(Ventanas)

Esta opción es útil solo para el manejo de las ventanas de texto que se pueden tener abiertas en la pantalla, por ejemplo se puede tener la respuesta de frecuencia de la planta G(jw) como también la tabla de valores de -1/N, ventanas que pueden ser cambiadas de tamaño fácilmente mediante el mouse. Las opciones que se tienen de manejar las ventanas son las mostradas en la figura A.7

*Resize/move Ctrl-F5.-* Cuando se escoge este caso, se puede cambiar el tamaño de las ventanas abiertas, como también moverlas mediante las flechas de cursor, pero es más simple manejar las pantallas por medio del mouse, ya que solamente colocando el mouse en el extremo inferior izquierdo de una ventana y presionando el botón izquierdo de este, se puede cambiar a gusto de uno el tamaño de la ventana.

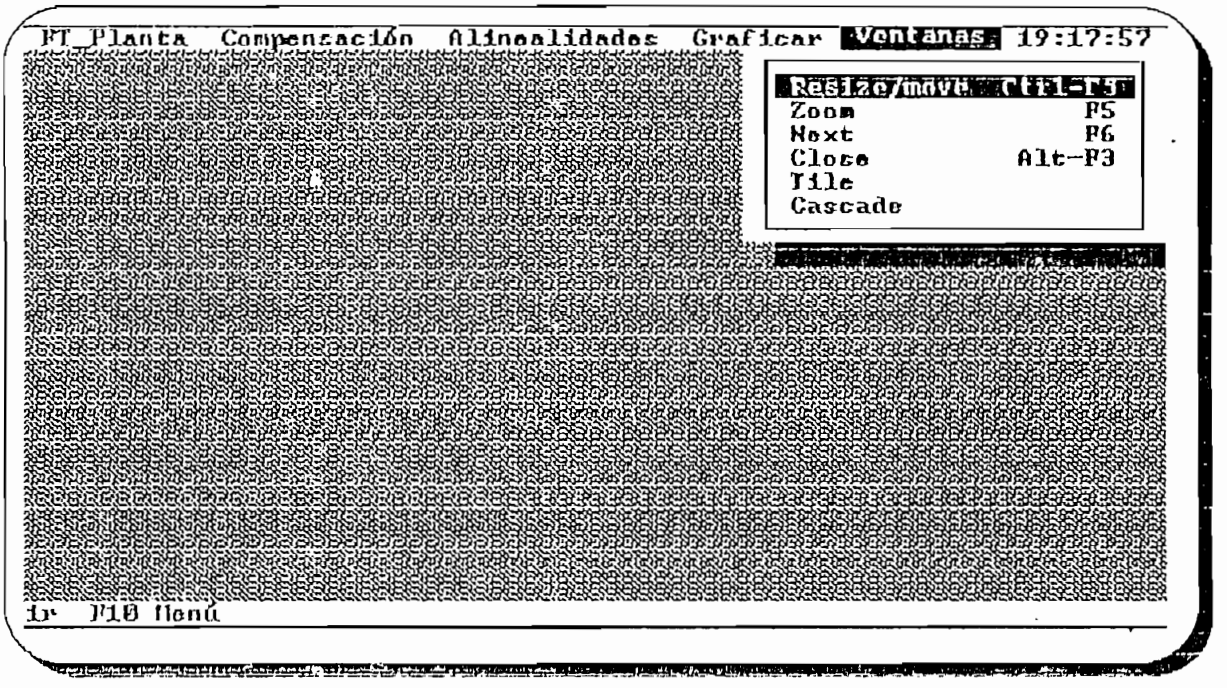



*Fig. A7 manejo de ventanas en el programa*

*Zoom F5.-* Se utiliza para agrandar al máximo la ventana abierta o para hacerla pequeña.

*Next F6.-* Cuando se tiene más de una ventana activa, se puede cambiar de una a otra con esta opción

*Close Alt-F3.-* Cierra la ventana que se encuentre activa

*Tile Cascada.-* Son opciones que permiten ordenar todas las ventanas abiertas en la pantalla una a continuación de otra o una atrás de otra respectivamente.

La forma como se distribuyen los archivos de datos cuando se graban en el programa son de la siguiente manera:

NOMBRE.CNL Es un archivo de utilidad solo para el programa, ya que es aquí donde se graba la función de transferencia de la planta

NOMBRE.Gc Este también es un archivo de utilidad solo para el programa, en el se guarda la compensación que se haya utilizado.

NOMBRE.FT Guarda la respuesta de frecuencia de la planta, es decir se tiene una tabla de datos de G(jW) ante variaciones de la frecuencia w.

NOMBRE.FTc Es similar al anterior, pero en este caso se guarda la respuesta de la planta con la compensación PID utilizada

NOMBRE.FD Guarda la tabla de valores de -1/N ante variaciones de la amplitud A, dependiendo del tipo de alinealidad que se tenga.

NOMBRE.TIM Se tiene una tabla de valores de la respuesta en el tiempo del sistema

# ANEXO B

## LISTADO DE PROGRAMAS

# ANEXO B

```
/*-------------------------------------------------------------*/
/*                                                             */
/*   PROGRAMA PARA ANALIZAR ESTABILIDAD                        */
/*   UTILIZANDO LA FUNCION DESCRIPTIVA                         */
/*                                                             */
/*   Wilmer Cárdenas Burbano 1994                              */
/*-------------------------------------------------------------*/

#define Uses_TApplication
#define Uses_MsgBox
#define Uses_TFileDialog
#define Uses_TSubMenu
#define Uses_TMenuBar
#define Uses_fpstream
#define Uses_TStatusLine
#define Uses_TStatusDef
#define Uses_TStatusItem
#define Uses_TKeys
#define Uses_TDeskTop

#include <tv.h>
#include <string.h>
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

#include "funcion.h"
#include "reloj.h"
#include "fileview.h"
#include "tvcmds.h"
#include "calc.h"

float numerador[12];
float denominador[12];
float realFT[200];
float imagFT[200];
float realFD[60];
float imagFD[60];
float realFTc[200];
float imagFTc[200];
float Yk[800];       //Vector para grafico en el tiempo
float T=0.1;        //Periodo para discretización

float Kp, Ki, Kd;

int longit_vect=200;  /*Esta en relación con RealFt e imagFT*/
int num_datosFD=60;   /*Esta en relación con RealFD e ImagFD*/

int gradnum=0, gradden=0;
float factor_dato=longit_vect/20;
                    /*factor que indica la variación de w hasta dato*/
                    /*factor_dato/longit_vect determina el incremento de w*/
int factor_eje=200;   /*factor para escala de ejes y gráfico*/
                    /*Si aumenta el factor el tamaño del gráfico aumenta*/
                    /*Cada subdivisión en el gráfico es 0.1*/
```

```
float inicioAFD;

int tipoNL=0, hayFT=0, hayGc=0;
float pto1, pto2, pto3, pto4;

float Wc, Ac1=-1, Ac2=-1;
int Ptocorte;


//FUNCION PRINCIPAL

int main(int argc, char **argv)
{
    TVAesfudes aesfudes(argc, argv);        //  Application Instance.
    aesfudes.run();
    return 0;
}



// CONSTRUCCION DE LAS APLICACIONES

TVAesfudes::TVAesfudes( int argc, char **argv ) :
    TProgInit( &TVAesfudes::initStatusLine,
               &TVAesfudes::initMenuBar,
               &TVAesfudes::initDeskTop )
{
    TView *w;
    char fileSpec[128];
    int len;

    TRect r = getExtent();
    r.a.x = r.b.x - 9;      r.b.y = r.a.y + 1;
    clock = new TClockView( r );
    insert(clock);

    r = getExtent();
    r.a.x = r.b.x - 13;     r.a.y = r.b.y - 1;
    heap = new THeapView( r );

    while (--argc > 0)
        {
        strcpy( fileSpec, *++argv );
        len = strlen( fileSpec );
        if( fileSpec[len-1] == '\\' )
            strcat( fileSpec, "*.*" );
        if( strchr( fileSpec, '*' ) || strchr( fileSpec, '?' ) )
            openFile( fileSpec );
        else
            {
            w = validView( new TFileWindow( fileSpec ) );
            if( w != 0 )
                deskTop->insert(w);
            }
        }

    system("m_inicio.exe");
    aboutDlgBox();
    disableCommand(cmImprimirFT);
    disableCommand(cmImprimirFD);
    disableCommand(cmImprimirGc);
    disableCommand(cmImprimirTime);
    disableCommand(cmGrabarFT);
    disableCommand(cmGrabarFD);
    disableCommand(cmGrabarGc);
    disableCommand(cmGrabarTime);
}
```

```
// CREAR LA LINEA DE ESTADO (STATUS LINE)

TStatusLine *TVAesfudes::initStatusLine( TRect r )
{
    r.a.y = r.b.y - 1;

    return (new TStatusLine( r,
      *new TStatusDef( 0, 0xFFFF ) +
        *new TStatusItem( "~Alt-S~ Salir", kbAltS, cmSalir ) +
        *new TStatusItem("~F10~ Menú", kbF10, cmMenu )
        )
    );
}



// writeView() función ( escribe una ventana de objeto a un archivo de recurso)

static void writeView(TView *p, void *strm)
{
   fpstream *s = (fpstream *) strm;
   if (p != TProgram::deskTop->last)
      *s << p;
}



// Función Tile

void TVAesfudes::tile()
{
    deskTop->tile( deskTop->getExtent() );
}



Boolean graphActive;
int driver;
int mode;
char *bgiPath;

Boolean TVAesfudes::graphicsStart(void)
{
    if ( graphActive == True )
        return(True);

    initgraph( &driver, &mode, bgiPath );

    if ( driver < 0 )
        {
        graphicsStop();
        return False;
        }
    else
        graphActive = True;
    return True;
}



void TVAesfudes::graphicsStop(void)
{
    if ( graphActive == True )
        closegraph();

    graphActive = False;
    TProgram::application->redraw();
}
```

```
// FUNCION PARA ABRIR ARCHIVOS

void TVAesfudes::openFile( char *fileSpec )
{
    TFileDialog *d= (TFileDialog *)validView(
    new TFileDialog(fileSpec, "Open a File", "~N~ame", fdOpenButton, 100 ));

    if( d != 0 && deskTop->execView( d ) != cmCancel )
        {
        char fileName[MAXPATH];
        d->getFileName( fileName );
        TView *w= validView( new TFileWindow( fileName ) );
        if( w != 0 )
            deskTop->insert(w);
    }
    destroy( d );
}



// FUNCION PARA REVISAR CAPACIDAD DE MEMORIA

void TVAesfudes::outOfMemory()
{
    messageBox( "No hay memoria disponible para completar la operación.",
      mfError | mfOKButton );
}



// PREPARACION DE LA PALETA DE COLORES

TPalette& TVAesfudes::getPalette() const
{
     static TPalette newcolor(cpColor,sizeof(cpColor)-1);
     static TPalette newblackwhite(cpBlackWhite,sizeof(cpBlackWhite)-1);
     static TPalette newmonochrome(cpMonochrome,sizeof(cpMonochrome)-1);
     static TPalette *palettes[] =
       {
         &newcolor, &newblackwhite, &newmonochrome
       };
     return *(palettes[appPalette]);

 }



// FUNCION PARA PONER EN FORMA TILE LA PANTALLA

static Boolean isTileable(TView *p, void * )
{
   if( (p->options & ofTileable) != 0)
       return True;
   else
       return False;
}


// FUNCION PARA ACTUALIZAR EL RELOJ EN LA PANTALLA

void TVAesfudes::idle()
{
    TProgram::idle();
    clock->update();
    heap->update();
    if (deskTop->firstThat(isTileable, 0) != 0 )
        {
```

```
        enableCommand(cmTile);
        enableCommand(cmCascade);
        }
    else
        {
        disableCommand(cmTile);
        disableCommand(cmCascade);
        }
}


// FUNCION PARA CERRAR VENTANAS

static void closeView(TView *p, void *p1)
{
    message(p, evCommand, cmClose, p1);
}


// loadDesktop() function

void TVAesfudes::loadDesktop(fpstream &s)
{
    TView  *p;

    if (deskTop->valid(cmClose))
        {
        deskTop->forEach(::closeView, 0);  // Clear the desktop
        do {
           s >> p;
           deskTop->insertBefore(validView(p), deskTop->last);
           }
           while (p != 0);
        }
}


// INICIALIZACION DE LA BARRA DE MENU

TMenuBar *TVAesfudes::initMenuBar(TRect r)
{
       TSubMenu& sub1 =
         *new TSubMenu( "~\360~", 0, hcNoContext) +
              *new TMenuItem("~T~itulo", cmAboutCmd, kbNoKey, hcNoContext)+
              *new TMenuItem("~C~alculadora", cmCalculadora, kbNoKey, hcNoContext);

       TSubMenu& sub2 =
         *new TSubMenu( "~F~ile", 0, hcNoContext) +
              *new TMenuItem( "~A~brir", cmOpenCmd, kbF3, hcNoContext, "F3" ) +
              *new TMenuItem(*new TSubMenu("~G~rabar", 0, hcNoContext)+
                    *new TMenuItem("Resp. Freq. ~F~T", cmGrabarFT, kbNoKey, hcNoContext)+
                    *new TMenuItem("Resp. Freq. ~G~c", cmGrabarGc, kbNoKey, hcNoContext)+
                    *new TMenuItem("Resp. Freq. F~D~", cmGrabarFD, kbNoKey, hcNoContext)+
                    *new TMenuItem("Resp. T~i~empo", cmGrabarTime, kbNoKey, hcNoContext)
                    )+
               newLine() +
              *new TMenuItem (*new TSubMenu("~I~mprimir", 0, hcNoContext)+
                        *new  TMenuItem("Resp.  de  Freq.  de  F~T~",cmImprimirFT,  kbNoKey,
hcNoContext) +
                        *new  TMenuItem("Resp.  de  Freq.  de  G~c~",cmImprimirGc,  kbNoKey,
hcNoContext) +
                        *new  TMenuItem("Resp.  de  Freq.  de  F~D~",cmImprimirFD,  kbNoKey,
hcNoContext) +
                   *new TMenuItem("Resp. T~i~empo",cmImprimirTime, kbNoKey, hcNoContext)
                   )+
               newLine() +
              *new TMenuItem( "~S~alir", cmSalir, kbAltS, hcNoContext, "Alt-S" );
```

```
        TSubMenu& sub3 =
          *new TSubMenu( "FT_~P~lanta", 0, hcNoContext) +
                *new TMenuItem( "FT ~N~ueva", cmDatNuev, kbNoKey, hcNoContext) +
                *new TMenuItem( "FT ~G~rabada", cmDatGrab, kbNoKey, hcNoContext)+
                 newLine() +
                *new TMenuItem( "~V~er la FT", cmVerFT, kbNoKey, hcNoContext);

        TSubMenu& sub4 =
          *new TSubMenu( "~C~ompensación", 0, hcNoContext) +
                *new TMenuItem( "Gc ~N~ueva", cmGcNuev, kbNoKey, hcNoContext) +
                *new TMenuItem( "Gc ~G~rabada", cmGcGrab, kbNoKey, hcNoContext)+
                 newLine() +
                *new TMenuItem( "~E~liminar la Gc", cmEliminarGc, kbNoKey, hcNoContext)+
                 newLine() +
                *new TMenuItem( "~V~er la Gc", cmVerGc, kbNoKey, hcNoContext);


        TSubMenu& sub5 =
          *new TSubMenu( "~A~linealidades", 0, hcNoContext)+
                *new TMenuItem(*new TSubMenu("~R~elés", 0, hcNoContext)+
                     *new TMenuItem("~I~deal", cmRele_Ideal, kbNoKey, hcNoContext)+
                     *new TMenuItem("con ~Z~ona Muerta", cmRele_ZM, kbNoKey, hcNoContext)+
                            *new  TMenuItem("con  ~P~endiente",  cmRele_Pendiente,  kbNoKey,
hcNoContext)+
                     *new TMenuItem("con Z.M. ~Y~ Pend.", cmRele_ZMyP, kbNoKey, hcNoContext)
                     )+

                *new TMenuItem(*new TSubMenu("~S~aturación", 0, hcNoContext)+
                     *new TMenuItem("~I~deal", cmSat_Ideal, kbNoKey, hcNoContext)+
                     *new TMenuItem("con ~Z~ona Muerta", cmSat_ZM, kbNoKey, hcNoContext)+
                     *new TMenuItem("con ~P~endiente", cmSat_Pendiente, kbNoKey, hcNoContext)+
                     *new TMenuItem("con Z.M. ~Y~ Pend.", cmSat_ZMyP, kbNoKey, hcNoContext)
                     )+

                *new TMenuItem(*new TSubMenu("~H~isteresis", 0, hcNoContext)+
                     *new TMenuItem("~I~deal", cmHist_Ideal, kbNoKey, hcNoContext)+
                     *new TMenuItem("con ~Z~ona Muerta", cmHist_ZM, kbNoKey, hcNoContext)+
                            *new  TMenuItem("con  ~P~endiente",  cmHist_Pendiente,  kbNoKey,
hcNoContext)+
                     *new TMenuItem("con Z.M. ~Y~ Pend.", cmHist_ZMyP, kbNoKey, hcNoContext)
                     )+
                 newLine() +
                *new TMenuItem( "~E~liminar la NL", cmEliminarFD, kbNoKey, hcNoContext)+
                 newLine() +
                *new TMenuItem( "~V~er la NL", cmVerFD, kbNoKey, hcNoContext);


        TSubMenu& sub6 =
          *new TSubMenu( "~G~raficar", 0, hcNoContext)+
                *new TMenuItem("Graf. -1/N y ~G~(jw)", cmGrafG, kbNoKey, hcNoContext)+
                *new TMenuItem("Graf. -1/N y G~c~(jw)", cmGrafGc, kbNoKey, hcNoContext)+
                newLine() +
                *new TMenuItem("Graf. ~R~esp. NL", cmGrafNL, kbNoKey, hcNoContext)+
                *new TMenuItem("Graf. Resp. ~T~iempo", cmGrafTiempo, kbNoKey, hcNoContext);

        TSubMenu& sub7 =
          *new TSubMenu( "V~e~ntanas", 0, hcNoContext) +
                *new TMenuItem( "~R~esize/move", cmResize, kbCtrlF5, hcNoContext, "Ctrl-F5" )
+
                *new TMenuItem( "~Z~oom", cmZoom, kbF5, hcNoContext, "F5" ) +
                *new TMenuItem( "~N~ext", cmNext, kbF6, hcNoContext, "F6" ) +
                *new TMenuItem( "~C~lose", cmClose, kbAltF3, hcNoContext, "Alt-F3" ) +
                *new TMenuItem( "~T~ile", cmTile, kbNoKey, hcNoContext) +
                *new TMenuItem( "C~a~scade", cmCascade, kbNoKey, hcNoContext);


        r.b.y =  r.a.y + 1;
        return (new TMenuBar( r, sub1+sub2+sub3+sub4+sub5+sub6+sub7) );
```

```
}


//  LAZO DE EVENTOS PARA DISTRIBUIR EL TRABAJO

void TVAesfudes::handleEvent(TEvent &event)
{
    int i;
    TApplication::handleEvent(event);

    if (event.what == evCommand)
    {
       switch (event.message.command)
           {
           case cmAboutCmd:            //  Dialog Box
                aboutDlgBox();
                break;

           case cmCalculadora:
                calculadora();
                break;

           case cmOpenCmd:             //  Abrir un archivo
                openFile("*.*");
                break;

           case cmTile:              //  "Tile" las ventanas
                tile();
                break;

           case cmCascade:           //  "Cascade" las ventanas
                cascade();
                break;

           case cmGrabarFT:          //Permite grabar los resultados
                grabar_resultadosFT();
                break;

           case cmGrabarFD:          //Permite grabar los resultados
                grabar_resultadosFD();
                break;

           case cmGrabarGc:          //Permite grabar los resultados
                grabar_resultadosGc();
                break;

           case cmGrabarTime:        //Permite grabar los resultados
                grabar_Time();        //de la respuesta en el tiempo
                break;


           case cmImprimirFT:         //  Imprime los resultados
                imprimirFT();         // de la función de Trasferencia
                break;

           case cmImprimirFD:         //  Imprime los resultados
                imprimirFD();         // de la función Descriptiva
                break;


           case cmImprimirGc:         //  Imprime los resultados
                imprimirGc();         // de FT con compensación
                break;

           case cmImprimirTime:        //  Imprime los resultados
                imprimirTime();        // de la respuesta en el tiempo
                break;
```

```
case cmDatNuev:             // Permite ingresar una FT
     i-datosnuevos();
     if(i--1)
       {
         i_numerador();
         i_denominador();
         disableCommand(cmImprimirGc);
         disableCommand(cmGrabarGc);
         coordenadasFT();
       }
     else
       {
         resume();
       }
     break;

case cmDatGrab:             // Permite escoger una FT grabada
     i-datosgrabados();
     if(i--1)
       {
         disableCommand(cmImprimirGc);
         disableCommand(cmGrabarGc);
         coordenadasFT();
       }
     break;

case cmVerFT:               // Permite ver la FT
     if(hayFT--1)
       verFT();
     else
       {
         puts("\a");
         messageBox("No se ha ingresado una Función de Transferencia",
          mfError | mfOKButton);
       }
     break;


case cmGcNuev:             // Permite ingresar una Gc
     if(hayFT--1)
       IngresoGc();
     else
       {
         puts("\a");
         messageBox("No se ha ingresado la Función de Transferencia de la planta",
          mfError | mfOKButton);
       }
     break;

case cmGcGrab:             // Permite escoger una Gc grabada
     if(hayFT--1)
       RecuperaGc();
     else
       {
         puts("\a");
         messageBox("No se ha ingresado la Función de Transferencia de la planta",
          mfError | mfOKButton);
       }
     break;

case cmVerGc:               // Permite ver la FT
     if(hayGc--1)
       verGc();
     else
       {
         puts("\a");
         messageBox("No se ha ingresado una Compensación",
          mfError | mfOKButton);
```

```
      }
    break;


case cmRele_Ideal:          // Permite escoger un Relé Ideal
    i=fig_rele_Ideal();
    if(i--1)
      rele_Ideal();
    else
      resume();
    break;

case cmRele_ZM:             // Permite escoger un Relé con
    i= fig_rele_ZM();       // Zona Muerta
    if(i--1)
      rele_ZM();
    else
      resume();
    break;

case cmRele_Pendiente:      // Permite escoger un Relé con
    i=fig_rele_Pend();      //  pendiente
    if(i--1)
      rele_Pend();
    else
      resume();
    break;

case cmRele_ZMyP:           // Permite escoger un Relé con
    i=fig_rele_ZMyP();      // Zona Muerta y Pendiente
    if(i--1)
      rele_ZMyP();
    else
      resume();
    break;


case cmSat_Ideal:           // Permite escoger una Saturación
    i=fig_sat_Ideal();      // Ideal
    if(i--1)
      sat_Ideal();
    else
      resume();
    break;

case cmSat_ZM:              // Permite escoger una Saturación
    i=fig_sat_ZM();         // con Zona Muerta
    if(i--1)
      sat_ZM();
    else
      resume();
    break;

case cmSat_Pendiente:       // Permite escoger una Saturación
    i=fig_sat_Pend();       // con Pendiente
    if(i--1)
      sat_Pend();
    else
      resume();
    break;

case cmSat_ZMyP:            // Permite escoger una Saturación
    i=fig_sat_ZMyP();       // con Zona Muerta y Pendiente
    if(i--1)
      sat_ZMyP();
    else
      resume();
    break;
```

```
case cmHist_Ideal:          // Permite escoger una Histéresis
     i=fig_hist_Ideal();    // Ideal
     if(i==1)
       hist_Ideal();
     else
       resume();
     break;

case cmHist_ZM:             // Permite escoger una Histéresis
     i=fig_hist_ZM();       // con Zona Muerta
     if(i==1)
       hist_ZM();
     else
       resume();
     break;

case cmHist_Pendiente:      // Permite escoger una Histéresis
     i=fig_hist_Pend();     // con Pendiente
     if(i==1)
       hist_Pend();
     else
       resume();
     break;

case cmHist_ZMyP:           // Permite escoger una Histéresis
     i=fig_hist_ZMyP();     // con Zona Muerta y Pendiente
     if(i==1)
       hist_ZMyP();
     else
       resume();
     break;


case cmVerFD:
     if(tipoNL==0)
       {
         puts("\a");
         messageBox("No se ha escogido una no linealidad",
          mfError | mfOKButton);
         i=0;
       }
     if(tipoNL==1)
       i=fig_rele_Ideal();
     if(tipoNL==2)
       i=fig_rele_ZM();
     if(tipoNL==3)
       i=fig_rele_Pend();
     if(tipoNL==4)
       i=fig_rele_ZMyP();
     if(tipoNL==5)
       i=fig_sat_Ideal();
     if(tipoNL==6)
       i=fig_sat_ZM();
     if(tipoNL==7)
       i=fig_sat_Pend();
     if(tipoNL==8)
       i=fig_sat_ZMyP();
     if(tipoNL==9)
       i=fig_hist_Ideal();
     if(tipoNL==10)
       i=fig_hist_ZM();
     if(tipoNL==11)
       i=fig_hist_Pend();
     if(tipoNL==12)
       i=fig_hist_ZMyP();
     if(i==1)
       verFD();
     break;
```

```
case cmEliminarGc:
     hayGc=0;
     disableCommand(cmImprimirGc);
     disableCommand(cmGrabarGc);
     break;

case cmEliminarFD:
     tipoNL=0;
     disableCommand(cmImprimirFD);
     disableCommand(cmGrabarFD);
     break;

case cmGrafG:              // Permite graficar -1/N y G(jw)
     if(hayFT==0 || tipoNL==0)
       {
         puts("\a");
         if(hayFT==0 && tipoNL==0)
           {
            messageBox("No hay Datos Ingresados",
             mfError | mfOKButton);
           }
         if(hayFT==0 && tipoNL!=0)
           {
            messageBox("No se ha ingresado una Función de Transferencia",
             mfError | mfOKButton);
           }
         if(hayFT!=0 && tipoNL==0)
           {
            messageBox("No se ha escogido una NoLinealidad",
             mfError | mfOKButton);
           }
       }
     else
       {
         puntosdecruce();
         i=graficar();
         if(i==1)
           {
            enableCommand(cmImprimirFT);
            enableCommand(cmImprimirFD);
            enableCommand(cmGrabarFT);
            enableCommand(cmGrabarFD);
            do
              {
                trazoejes();
                graficFT();
                i=graficFD();
              }while(i==1);
            graphicsStop();
            resume();
           }
         else
           resume();
       }
     break;


case cmGrafGc:             // Permite graficar -1/N y Gc(jw)
     if(hayGc==0 || tipoNL==0)
       {
         puts("\a");
         if(hayGc==0 && tipoNL==0)
           {
            messageBox("No hay Datos Ingresados",
             mfError | mfOKButton);
           }
         if(hayGc==0 && tipoNL!=0)
```

```
            {
             messageBox("No se ha ingresado una Compensación",
              mfError | mfOKButton);
            }
          if(hayGc!=0 && tipoNL==0)
            {
             messageBox("No se ha escogido una NoLinealidad",
              mfError | mfOKButton);
            }
        }
      else
        {
          puntosdecruceGc();
          i=graficar();
          if(i==1)
            {
             enableCommand(cmImprimirGc);
             enableCommand(cmImprimirFD);
             enableCommand(cmGrabarGc);
             enableCommand(cmGrabarFD);
             do
               {
                 trazoejes();
                 graficGc();
                 i=graficFD();
               }while(i==1);
             graphicsStop();
             resume();
            }
          else
            resume();
        }
      break;



case cmGrafNL:
  if(tipoNL==0)
    {
      puts("\a");
      messageBox("No se ha escogido una NoLinealidad",
      mfError | mfOKButton);
    }
  else
    {
      i=graficar();
      if(i==1)
        {
          grafNL();
          getch();
          graphicsStop();
          resume();
        }
      else
      resume();
    }
  break;



case cmGrafTiempo:
     if(hayFT==0)
       {
         puts("\a");
         messageBox("No se ha ingresado una Función de Transferencia",
           mfError | mfOKButton);
       }
```

```
                {
                  i-graficar();
                  if(i--1)
                    {
                      calculoTiempo();
                      do
                        {
                          trazoejesTiempo();
                          i-graficTiempo();
                        }while(i--1);
                      graphicsStop();
                      enableCommand(cmImprimirTime);
                      enableCommand(cmGrabarTime);
                      resume();
                    }
                  else
                    resume();
                }
            break;


          case cmSalir:
               system("m_final.exe");
               endModal(cmQuit);


          default:
               return;

          }
        clearEvent (event);
        }
}



// Función "Cascade"

void TVAesfudes::cascade()
{
    deskTop->cascade( deskTop->getExtent() );
}


void TVAesfudes::calculadora()
{
   TCalculator *calc = (TCalculator *) validView(new TCalculator);
   if(calc != 0)
      deskTop -> insert(calc);
}
```

```
/*-------------------------------------------------------*/
/*                                                       */
/*   PROGRAMA PARA ANALIZAR ESTABILIDAD                  */
/*   UTILIZANDO LA FUNCION DESCRIPTIVA                   */
/*                                                       */
/*    PRESENTACION Y SALIDA DE DATOS                     */
/*                                                       */
/*   Wilmer Cárdenas Burbano 1994                        */
/*-------------------------------------------------------*/


#define Uses_MsgBox
#define Uses_TApplication
#define Uses_TStaticText
#define Uses_TButton
#define Uses_TColorDialog
#define Uses_TDeskTop
#define Uses_TInputLine

#define pi 3.141592

#include <tv.h>
#include <stdio.h>
#include <conio.h>
#include <bios.h>
#include <math.h>

#include "funcion.h"


extern float numerador[];
extern float denominador[];
extern float realFT[];
extern float imagFT[];
extern float realFD[];
extern float imagFD[];
extern float realFTc[];
extern float imagFTc[];

extern float Yk[];

extern int gradnum, gradden, longit_vect, num_datosFD;
extern float factor_dato, inicioAFD;
extern int tipoNL, hayGc;
extern float pto1, pto2, pto3, pto4;

extern float Kp, Kd, Ki;
extern float T;


// FUNCION DE PRESENTACION

void TVAesfudes::aboutDlgBox()
{
    TDialog *aboutBox = new TDialog(TRect(0, 0, 39, 14), "Titulo");
    aboutBox->insert(
      new TStaticText(TRect(9, 2, 32, 13),
        "\003PROGRAMA SEMIGRAFICO\n"
        "\003PARA ANALIZAR\n"
        "\003ESTABILIDAD\n"
        "\003USANDO LA FUNCION\n"
        "\003DESCRIPTIVA\n\003\n"
        "\003AUTOR\n"
        "\003Wilmer Cárdenas Burbano\n"
        "\0031994"
        )
      );
```

```
    aboutBox->insert(
      new TButton(TRect(14, 11, 25, 13), " OK", cmOK, bfDefault)
      );

    aboutBox->options |= ofCentered;
    deskTop->execView(aboutBox);
    destroy( aboutBox );
}



 /*FUNCION PARA GRABAR LOS RESULTADOS EN UN ARCHIVO*/
int TVAesfudes::grabar_resultadosFT()
  {
    char archivo[30];
    int i, j;
    float dato, w, inicio, A, Mod, Ang;
    FILE *fpout;

    for(i=0;i<30;i++)
      archivo[i]='\0';

    i=0;
    TDialog *grabar = new TDialog(TRect(0, 0, 40, 13), "GRABAR");
    grabar->insert(
      new TStaticText(TRect(5, 2, 35, 8),
        "\003GRABAR LA RESPUESTA DE\n"
        "\003FRECUENCIA DE LA PLANTA\n\003\n"
        "\003Ingrese el nombre\n"
        "\003del archivo de salida\n"
        "\003(sin extensión)"
        )
      );
    grabar->insert(new TInputLine(TRect(10,8,30,9),30));
    grabar->insert(
      new TButton(TRect(7, 10, 17, 12), "~O~K", cmOK, bfDefault)
      );
    grabar->insert(
      new TButton(TRect(23, 10, 33, 12), "~C~ancel", cmCancel, bfNormal)
      );
    grabar->options |= ofCentered;

    grabar->setData(archivo);

    ushort control=deskTop->execView(grabar);
    if(control == cmOK)
      {
        i=1;
        grabar->getData(archivo);
      }
    destroy(grabar);

    if(i==1)
      {
        for(i=1;i<30;i++)
          {
            if(archivo[i]=='\0')
              {
                j=i;
                i=30;
              }
          }

        archivo[j]='.';
        archivo[j+1]='c';
        archivo[j+2]='n';
        archivo[j+3]='l';
```

```
i=1;
if((fpout=fopen(archivo,"r"))!=NULL)
  {
    fclose(fpout);
    puts("\a");
    i=0;
    TDialog *warning = new TDialog(TRect(0, 0, 40, 11), "WARNING");
      warning->insert(
      new TStaticText(TRect(5, 2, 35, 7),
      "\003EL ARCHIVO INDICADO\n"
      "\003YA EXISTE\n\003\n"
      "\003Desea Sobreescribir\n"
      "\003en el archivo\n"
      )
    );
    warning->insert(
      new TButton(TRect(7, 8, 17, 10), "~S~i", cmOK, bfDefault)
     );
    warning->insert(
      new TButton(TRect(23, 8, 33, 10), "~N~o", cmCancel, bfNormal)
    );
    warning->options |= ofCentered;

    ushort controle=deskTop->execView(warning);
    if(controle == cmOK)
      i=1;
    destroy(warning);
  }

if(i==1)
  {
    if((fpout=fopen(archivo,"w"))==NULL)
      {
       puts("\a");
       messageBox("No se puede escribir en el archivo",
        mfError | mfOKButton);
       return 0;
      }

    fprintf(fpout,"%d\n%d",gradnum,gradden);

    for(i=0;i<=gradnum;i++)
    fprintf(fpout,"\n%f",numerador[i]);

    for(i=0;i<=gradden;i++)
    fprintf(fpout,"\n%f",denominador[i]);

    fclose(fpout);


    /*FUNCION GUARDAR DATOS DE LA FT EN UN ARCHIVO*/
    archivo[j+1]='f';
    archivo[j+2]='t';
    archivo[j+3]='\0';

    if((fpout=fopen(archivo,"w"))==NULL)
      {
       puts("\a");
       messageBox("No se puede escribir en el archivo",
        mfError | mfOKButton);
       return 0;
      }

    fprintf(fpout,
          "         PROGRAMA SEMIGRAFICO PARA ANALIZAR ESTABLIDAD");
    fprintf(fpout,"\n                  USANDO LA FUNCION DESCRIPTIVA");

    fprintf(fpout,"\n\n         FUNCION DE TRANSFERENCIA");
    fprintf(fpout,"\n\n    Grado del Numerador=   %d",gradnum);
```

```
        fprintf(fpout,"\n    Grado del Denominador- %d",gradden);

        fprintf(fpout,"\n\n    NUMERADOR");
        for(i=0;i<=gradnum;i++)
          fprintf(fpout,"\n      a%d- %f",i,numerador[i]);
        fprintf(fpout,"\n\n    DENOMINADOR");
        for(i=0;i<=gradden;i++)
          fprintf(fpout,"\n      b%d- %f",i,denominador[i]);
        fprintf(fpout,"\n\n RESPUESTA DE FRECUENCIA DE LA FUNCION DE TRANSFERENCIA\n");

        inicio=factor_dato/(float)longit_vect;
        fprintf(fpout, "   %s        %s       %s     %s     %s\n",
             "W", "Real FT.", "Imag. FT.", "Módulo FT", "Angulo FT");

        for(i=0; i<longit_vect; i++)
          {
           if(i<80)
             w=inicio*(i+1);
           if(i>=80 && i<=190)
             w=0.4*i-27.6;
           if(i>190)
             w=2*i-331.6;

           Mod = sqrt((double)realFT[i]*realFT[i]+imagFT[i]*imagFT[i]);
           if(realFT[i]!=0)
             {
               Ang = atan((double)imagFT[i]/realFT[i])*180/pi;
               if(realFT[i]<0)
                 Ang = 180 + Ang;
               if(realFT[i]>0 && imagFT[i]<0)
                 Ang = 360 + Ang;
             }
           else
             {
               if(imagFT[i]<0)
                 Ang = 270;
               else
                 Ang = 90;
             }

           fprintf(fpout,"%6.2f     %9.3e     %9.3e     %9.3e     %6.3f\n",
                w, realFT[i], imagFT[i], Mod, Ang);
          }
        fclose(fpout);
       }
     }
   return 0;
}



int TVAesfudes::grabar_resultadosFD()
  {
    char archivo[30];
    int i, j;
    float dato, w, inicio, A, Mod, Ang;
    FILE *fpout;

    for(i=0;i<30;i++)
      archivo[i]='\0';

    i=0;
    TDialog *grabar = new TDialog(TRect(0, 0, 40, 13), "GRABAR");
    grabar->insert(
      new TStaticText(TRect(5, 2, 35, 8),
        "\003GRABAR LA RESPUESTA DE\n"
        "\003FRECUENCIA DE LA ALINEALIDAD\n\003\n"
        "\003ingrese el nombre\n"
```

```
    "\003del archivo de salida\n"
    "\003(sin extensión)"
    )
  );
grabar->insert(new TInputLine(TRect(10,8,30,9),30));
grabar->insert(
  new TButton(TRect(7, 10, 17, 12), "~O~K", cmOK, bfDefault)
  );
grabar->insert(
  new TButton(TRect(23, 10, 33, 12), "~C~ancel", cmCancel, bfNormal)
  );
grabar->options |= ofCentered;

grabar->setData(archivo);

ushort control=deskTop->execView(grabar);
if(control == cmOK)
  {
    i=1;
    grabar->getData(archivo);
  }
destroy(grabar);

if(i==1)
  {
    for(i=1;i<30;i++)
      {
        if(archivo[i]=='\0')
          {
           j=i;
           i=30;
          }
      }

    archivo[j]='.';
    archivo[j+1]='f';
    archivo[j+2]='d';
    archivo[j+3]='\0';

    i=1;
    if((fpout=fopen(archivo,"r"))!=NULL)
      {
        fclose(fpout);
        puts("\a");
        i=0;
        TDialog *warning = new TDialog(TRect(0, 0, 40, 11), "WARNING");
          warning->insert(
          new TStaticText(TRect(5, 2, 35, 7),
          "\003EL ARCHIVO INDICADO\n"
          "\003YA EXISTE\n\003\n"
          "\003Desea Sobreescribir\n"
          "\003en el archivo\n"
          )
        );
        warning->insert(
          new TButton(TRect(7, 8, 17, 10), "~S~i", cmOK, bfDefault)
         );
        warning->insert(
          new TButton(TRect(23, 8, 33, 10), "~N~o", cmCancel, bfNormal)
        );
        warning->options |= ofCentered;

        ushort controle=deskTop->execView(warning);
        if(controle == cmOK)
          i=1;

        destroy(warning);
      }
```

```
if(i--1)
  {
    if((fpout=fopen(archivo,"w"))--NULL)
      {
       puts("\a");
       messageBox("No se puede escribir en el archivo",
        mfError | mfOKButton);
       return 0;
      }


    /*FUNCION PARA GUARDAR DATOS DE LA FUNCION DESCRIPTIVA EN UN ARCHIVO*/

    fprintf(fpout,
        "\n\n        PROGRAMA SEMIGRAFICO PARA ANALIZAR ESTABLIDAD");
    fprintf(fpout,"\n                USANDO LA FUNCION DESCRIPTIVA");

    fprintf(fpout,"\n\n        FUNCION DESCRIPTIVA\n");
    if(tipoNL--0)
      fprintf(fpout,"        NO HAY ALINEALIDAD ESCOGIDA");
    if(tipoNL--1)
      {
       fprintf(fpout,"    RELE IDEAL\n");
       fprintf(fpout,"      U - %f\n",pto1);
      }
    if(tipoNL--2)
      {
       fprintf(fpout,"    RELE CON ZONA MUERTA\n");
       fprintf(fpout,"      U - %f\n",pto1);
       fprintf(fpout,"      D = %f\n",pto2);
      }
    if(tipoNL--3)
      {
       fprintf(fpout,"    RELE CON PENDIENTE\n");
       fprintf(fpout,"      U - %f\n",pto1);
       fprintf(fpout,"      K - %f\n",pto2);
      }
    if(tipoNL--4)
      {
       fprintf(fpout,"    RELE CON ZONA MUERTA Y PENDIENTE\n");
       fprintf(fpout,"      U - %f\n",pto1);
       fprintf(fpout,"      D - %f\n",pto2);
       fprintf(fpout,"      K - %f\n",pto3);
      }
    if(tipoNL--5)
      {
       fprintf(fpout,"    SATURACION IDEAL\n");
       fprintf(fpout,"      U - %f\n",pto1);
       fprintf(fpout,"      D - %f\n",pto2);
      }
    if(tipoNL--6)
      {
       fprintf(fpout,"    SATURACION CON ZONA MUERTA\n");
       fprintf(fpout,"      U - %f\n",pto1);
       fprintf(fpout,"      D1- %f\n",pto2);
       fprintf(fpout,"      D2- %f\n",pto3);
      }
    if(tipoNL--7)
      {
       fprintf(fpout,"    SATURACION CON PENDIETE\n");
       fprintf(fpout,"      U - %f\n",pto1);
       fprintf(fpout,"      D - %f\n",pto2);
       fprintf(fpout,"      K - %f\n",pto3);
      }
    if(tipoNL--8)
      {
       fprintf(fpout,"    SATURACION CON ZONA MUERTA Y PENDIENTE\n");
       fprintf(fpout,"      U - %f\n",pto1);
       fprintf(fpout,"      D1- %f\n",pto2);
```

```
        fprintf(fpout,"      D2= %f\n",pto3);
        fprintf(fpout,"      K = %f\n",pto4);
      }
    if(tipoNL==9)
      {
        fprintf(fpout,"    HISTERESIS IDEAL\n");
        fprintf(fpout,"      U = %f\n",pto1);
        fprintf(fpout,"      D = %f\n",pto2);
      }
    if(tipoNL==10)
      {
        fprintf(fpout,"    HISTERESIS CON ZONA MUERTA\n");
        fprintf(fpout,"      U = %f\n",pto1);
        fprintf(fpout,"      D1= %f\n",pto2);
        fprintf(fpout,"      D2= %f\n",pto3);
      }
    if(tipoNL==11)
      {
        fprintf(fpout,"    HISTERESIS CON PENDIENTE\n");
        fprintf(fpout,"      U = %f\n",pto1);
        fprintf(fpout,"      D1= %f\n",pto2);
        fprintf(fpout,"      D2= %f\n",pto3);
      }
    if(tipoNL==12)
      {
        fprintf(fpout,"    HISTERESIS CON ZONA MUERTA Y PENDIENTE\n");
        fprintf(fpout,"      U = %f\n",pto1);
        fprintf(fpout,"      D1= %f\n",pto2);
        fprintf(fpout,"      D2= %f\n",pto3);
        fprintf(fpout,"      D3= %f\n",pto4);
      }
    fprintf(fpout,"\n\n RESPUESTA DE FRECUENCIA DE LA FUNCION DESCRIPTIVA\n");

    if(tipoNL!=0)
      {
        fprintf(fpout, "     %s       %s     %s     %s     %s\n",
              "A","Real FD.", "Imag. FD.", "Módulo FD", "Angulo FD");
        for(i=0; i<num_datosFD; i++)
          {
            if(i<=20)
              A=0.3*i+inicioAFD;
            else
              A=2*i-34+inicioAFD;

            Mod = sqrt((double)realFD[i]*realFD[i]+imagFD[i]*imagFD[i]);
            if(realFD[i]!=0)
              {
                Ang = atan((double)imagFD[i]/realFD[i])*180/pi;
                if(realFD[i]<0)
                  Ang = 180 + Ang;
                if(realFD[i]>0 && imagFD[i]<0)
                  Ang = 360 + Ang;
              }
            else
              {
                if(imagFD[i]<0)
                  Ang = 270;
                else
                   Ang = 90;
              }

            fprintf(fpout,"%6.3f     %9.3e     %9.3e     %9.3e     %6.3f\n",
                  A, realFD[i], imagFD[i], Mod, Ang);
          }
      }
    fclose(fpout);
  }
}
```

```
}


int TVAesfudes::grabar_resultadosGc()
{
  char archivo[30];
  int i, j;
  float dato, w, inicio, A, Mod, Ang;
  FILE *fpout;

  for(i=0;i<30;i++)
    archivo[i]='\0';

  i=0;
  TDialog *grabar = new TDialog(TRect(0, 0, 40, 13), "GRABAR");
  grabar->insert(
    new TStaticText(TRect(5, 2, 35, 8),
      "\003GRABAR LA RESPUESTA DE FRECUENCIA\n"
      "\003DE LA PLANTA CON COMPENSACION\n\003\n"
      "\003Ingrese el nombre\n"
      "\003del archivo de salida\n"
      "\003(sin extensión)"
      )
    );
  grabar->insert(new TInputLine(TRect(10,8,30,9),30));
  grabar->insert(
    new TButton(TRect(7, 10, 17, 12), "~O~K", cmOK, bfDefault)
    );
  grabar->insert(
    new TButton(TRect(23, 10, 33, 12), "~C~ancel", cmCancel, bfNormal)
    );
  grabar->options |= ofCentered;

  grabar->setData(archivo);

  ushort control=deskTop->execView(grabar);
  if(control == cmOK)
    {
      i=1;
      grabar->getData(archivo);
    }
  destroy(grabar);

  if(i==1)
    {
      for(i=1;i<30;i++)
        {
          if(archivo[i]=='\0')
            {
              j=i;
              i=30;
            }
        }

      archivo[j]='.';
      archivo[j+1]='g';
      archivo[j+2]='c';
      archivo[j+3]='\0';

      i=1;
      if((fpout=fopen(archivo,"r"))!=NULL)
        {
          fclose(fpout);
          puts("\a");
          i=0;
          TDialog *warning = new TDialog(TRect(0, 0, 40, 11), "WARNING");
```

```
      warning->insert(
      new TStaticText(TRect(5, 2, 35, 7),
      "\003EL ARCHIVO INDICADO\n"
      "\003YA EXISTE\n\003\n"
      "\003Desea Sobreescribir\n"
      "\003en el archivo\n"
      )
    );
    warning->insert(
      new TButton(TRect(7, 8, 17, 10), "~S~i", cmOK, bfDefault)
     );
    warning->insert(
      new TButton(TRect(23, 8, 33, 10), "~N~o", cmCancel, bfNormal)
    );
    warning->options |= ofCentered;

    ushort controle=deskTop->execView(warning);
    if(controle == cmOK)
      i=1;

    destroy(warning);
  }

if(i==1)
  {
    if((fpout=fopen(archivo,"w"))==NULL)
      {
        puts("\a");
        messageBox("No se puede escribir en el archivo",
         mfError | mfOKButton);
        return 0;
      }

    fprintf(fpout,"%f\n%f\n%f",Kp, Kd, Ki);
    fclose(fpout);


    /*FUNCION GUARDAR DATOS DE LA FTc EN UN ARCHIVO*/
    archivo[j+1]='f';
    archivo[j+2]='t';
    archivo[j+3]='c';

    if((fpout=fopen(archivo,"w"))==NULL)
      {
        puts("\a");
        messageBox("No se puede escribir en el archivo",
         mfError | mfOKButton);
        return 0;
      }

    fprintf(fpout,
      "        PROGRAMA SEMIGRAFICO PARA ANALIZAR ESTABLIDAD");
    fprintf(fpout,"\n                USANDO LA FUNCION DESCRIPTIVA");

    fprintf(fpout,"\n\n        FUNCION DE TRANSFERENCIA");
    fprintf(fpout,"\n\n    Grado del Numerador-   %d",gradnum);
    fprintf(fpout,"\n    Grado del Denominador- %d",gradden);

    fprintf(fpout,"\n\n    NUMERADOR");
    for(i=0;i<=gradnum;i++)
      fprintf(fpout,"\n      a%d- %f",i,numerador[i]);
    fprintf(fpout,"\n\n    DENOMINADOR");
    for(i=0;i<=gradden;i++)
      fprintf(fpout,"\n      b%d- %f",i,denominador[i]);


    fprintf(fpout,"\n\n         COMPENSACION UTILIZADA");
    fprintf(fpout,"\n\n    Kp - %f",Kp);
    fprintf(fpout,"\n    Kd - %f",Kd);
```

```
        fprintf(fpout,"\n    Ki - %f",Ki);

        fprintf(fpout,"\n\n RESPUESTA DE FRECUENCIA DE LA FT. CON LA COMPENSACION\n");

        inicio=factor_dato/(float)longit_vect;
        fprintf(fpout, "   %s        %s       %s     %s     %s\n",
         "W", "Real FTc.", "Imag. FTc.", "Módulo FTc", "Angulo FTc");

        for(i=0; i<longit_vect; i++)
          {
            if(i<80)
             w=inicio*(i+1);
            if(i>=80 && i<=190)
             w=0.4*i-27.6;
            if(i>190)
             w=2*i-331.6;

            Mod = sqrt((double)realFTc[i]*realFTc[i]+imagFTc[i]*imagFTc[i]);
            if(realFTc[i]!=0)
             {
               Ang = atan((double)imagFTc[i]/realFTc[i])*180/pi;
               if(realFTc[i]<0)
                 Ang = 180 + Ang;
               if(realFTc[i]>0 && imagFTc[i]<0)
                 Ang = 360 + Ang;
             }
            else
             {
               if(imagFTc[i]<0)
                 Ang = 270;
               else
                 Ang = 90;
             }

            fprintf(fpout,"%6.2f     %9.3e     %9.3e     %9.3e     %6.3f\n",
                w, realFTc[i], imagFTc[i], Mod, Ang);
          }
        fclose(fpout);
      }
    }
  return 0;
 }



int TVAesfudes::grabar_Time()
{
  char archivo[30];
  int i, j;
  FILE *fpout;

  for(i=0;i<30;i++)
    archivo[i]='\0';

  i=0;
  TDialog *grabar = new TDialog(TRect(0, 0, 40, 13), "GRABAR");
  grabar->insert(
    new TStaticText(TRect(5, 2, 35, 8),
    "\003GRABAR LA RESPUESTA DEL\n"
    "\003SISTEMA EN EL TIEMPO\n\003\n"
    "\003Ingrese el nombre\n"
    "\003del archivo de salida\n"
    "\003(sin extensión)"
      )
    );
  grabar->insert(new TInputLine(TRect(10,8,30,9),30));
  grabar->insert(
    new TButton(TRect(7, 10, 17, 12), "~O~K", cmOK, bfDefault)
```

```
  );
grabar->insert(
  new TButton(TRect(23, 10, 33, 12), "~C~ancel", cmCancel, bfNormal)
  );
grabar->options |= ofCentered;

grabar->setData(archivo);

ushort control=deskTop->execView(grabar);
if(control == cmOK)
  {
    i=1;
    grabar->getData(archivo);
  }
destroy(grabar);

if(i==1)
  {
    for(i=1;i<30;i++)
     {
       if(archivo[i]=='\0')
         {
           j=i;
           i=30;
         }
     }

    archivo[j]='.';
    archivo[j+1]='t';
    archivo[j+2]='i';
    archivo[j+3]='m';

    i=1;
    if((fpout=fopen(archivo,"r"))!=NULL)
     {
       fclose(fpout);
       puts("\a");
       i=0;
       TDialog *warning = new TDialog(TRect(0, 0, 40, 11), "WARNING");
         warning->insert(
         new TStaticText(TRect(5, 2, 35, 7),
         "\003EL ARCHIVO INDICADO\n"
         "\003YA EXISTE\n\003\n"
         "\003Desea Sobreescribir\n"
         "\003en el archivo\n"
         )
       );
       warning->insert(
         new TButton(TRect(7, 8, 17, 10), "~S~i", cmOK, bfDefault)
        );
       warning->insert(
         new TButton(TRect(23, 8, 33, 10), "~N~o", cmCancel, bfNormal)
       );
       warning->options |= ofCentered;

       ushort controle=deskTop->execView(warning);
       if(controle == cmOK)
         i=1;

       destroy(warning);
     }

    if(i==1)
     {
       if((fpout=fopen(archivo,"w"))==NULL)
         {
           puts("\a");
           messageBox("No se puede escribir en el archivo",
```

```
    return 0;
  }


/*FUNCION GUARDAR DATOS DE LA RESPUESTA EN EL TIEMPO EN UN ARCHIVO*/
fprintf(fpout,
   "        PROGRAMA SEMIGRAFICO PARA ANALIZAR ESTABLIDAD");
fprintf(fpout,"\n                USANDO LA FUNCION DESCRIPTIVA");

fprintf(fpout,"\n\n         FUNCION DE TRANSFERENCIA");
fprintf(fpout,"\n\n    Grado del Numerador=   %d",gradnum);
fprintf(fpout,"\n    Grado del Denominador= %d",gradden);

fprintf(fpout,"\n\n    NUMERADOR");
for(i=0;i<=gradnum;i++)
  fprintf(fpout,"\n      a%d= %f",i,numerador[i]);
fprintf(fpout,"\n\n    DENOMINADOR");
for(i=0;i<=gradden;i++)
  fprintf(fpout,"\n      b%d= %f",i,denominador[i]);

if(hayGc!=0)
  {
    fprintf(fpout,"\n\n         COMPENSACION UTILIZADA");
    fprintf(fpout,"\n\n    Kp = %f",Kp);
    fprintf(fpout,"\n    Kd = %f",Kd);
    fprintf(fpout,"\n    Ki = %f",Ki);
  }

if(tipoNL!=0)
  {
    fprintf(fpout,"\n\n        FUNCION DESCRIPTIVA\n");
    if(tipoNL==1)
      {
        fprintf(fpout,"    RELE IDEAL\n");
        fprintf(fpout,"      U = %f\n",pto1);
      }
    if(tipoNL==2)
      {
        fprintf(fpout,"    RELE CON ZONA MUERTA\n");
        fprintf(fpout,"      U = %f\n",pto1);
        fprintf(fpout,"      D = %f\n",pto2);
      }
    if(tipoNL==3)
      {
        fprintf(fpout,"    RELE CON PENDIENTE\n");
        fprintf(fpout,"      U = %f\n",pto1);
        fprintf(fpout,"      K = %f\n",pto2);
      }
    if(tipoNL==4)
      {
        fprintf(fpout,"    RELE CON ZONA MUERTA Y PENDIENTE\n");
        fprintf(fpout,"      U = %f\n",pto1);
        fprintf(fpout,"      D = %f\n",pto2);
        fprintf(fpout,"      K = %f\n",pto3);
      }
    if(tipoNL==5)
      {
        fprintf(fpout,"    SATURACION IDEAL\n");
        fprintf(fpout,"      U = %f\n",pto1);
        fprintf(fpout,"      D = %f\n",pto2);
      }
    if(tipoNL==6)
      {
        fprintf(fpout,"    SATURACION CON ZONA MUERTA\n");
        fprintf(fpout,"      U = %f\n",pto1);
        fprintf(fpout,"      D1= %f\n",pto2);
        fprintf(fpout,"      D2= %f\n",pto3);
      }
    if(tipoNL==7)
```

```
            {
              fprintf(fpout,"    SATURACION CON PENDIETE\n");
              fprintf(fpout,"      U = %f\n",pto1);
              fprintf(fpout,"      D = %f\n",pto2);
              fprintf(fpout,"      K = %f\n",pto3);
            }
          if(tipoNL==8)
            {
              fprintf(fpout,"    SATURACION CON ZONA MUERTA Y PENDIENTE\n");
              fprintf(fpout,"      U = %f\n",pto1);
              fprintf(fpout,"      D1= %f\n",pto2);
              fprintf(fpout,"      D2= %f\n",pto3);
              fprintf(fpout,"      K = %f\n",pto4);
            }
          if(tipoNL==9)
            {
              fprintf(fpout,"    HISTERESIS IDEAL\n");
              fprintf(fpout,"      U = %f\n",pto1);
              fprintf(fpout,"      D = %f\n",pto2);
            }
          if(tipoNL==10)
            {
              fprintf(fpout,"    HISTERESIS CON ZONA MUERTA\n");
              fprintf(fpout,"      U = %f\n",pto1);
              fprintf(fpout,"      D1= %f\n",pto2);
              fprintf(fpout,"      D2= %f\n",pto3);
            }
          if(tipoNL==11)
            {
              fprintf(fpout,"    HISTERESIS CON PENDIENTE\n");
              fprintf(fpout,"      U = %f\n",pto1);
              fprintf(fpout,"      D1= %f\n",pto2);
              fprintf(fpout,"      D2= %f\n",pto3);
            }
          if(tipoNL==12)
            {
              fprintf(fpout,"    HISTERESIS CON ZONA MUERTA Y PENDIENTE\n");
              fprintf(fpout,"      U = %f\n",pto1);
              fprintf(fpout,"      D1= %f\n",pto2);
              fprintf(fpout,"      D2= %f\n",pto3);
              fprintf(fpout,"      D3= %f\n",pto4);
            }
        }

      fprintf(fpout,"\n\n RESPUESTA EN EL TIEMPO\n");
      fprintf(fpout, "   %s        %s\n", "t[s]", "Yk");

      for(i=0; i<800; i++)
        {
          fprintf(fpout,"%6.4f     %9.3e\n", T*25*i, Yk[i]);
        }
      fclose(fpout);
    }
  }
 return 0;
}



//I M P R I M I R

/* FUNCION PARA IMPRIMIR LOS RESULTADOS DE LA FUNCION DE TRANSFERENCIA*/
int TVAesfudes::imprimirF1()
 {
   int i, status;
   float dato, W, inicio, Mod, Ang;
   FILE *fpout;
```

```
i=0;
TDialog *imprimir = new TDialog(TRect(0, 0, 40, 12), "Imprimir");

imprimir->insert(
  new TStaticText(TRect(5, 2, 35, 8),
    "\003IMPRESION DE LA RESPUESTA\n"
    "\003DE FRECUENCIA DE LA PLANTA\n\003\n"
    "\003Asegurese de que\n"
    "\003La impresora este Lista?"
    )
  );

imprimir->insert(
  new TButton(TRect(7, 9, 17, 11), "~O~K", cmOK, bfNormal)
  );
imprimir->insert(
  new TButton(TRect(23, 9, 33, 11), "~C~ancel", cmCancel, bfDefault)
  );
imprimir->options |= ofCentered;

ushort control=deskTop->execView(imprimir);
if(control == cmOK)
  i=1;

destroy( imprimir);

if (i==1)
  {
   status = biosprint(_PRINTER_STATUS, 0, 0);
   if (status & 0x01)
     {
       puts("\a");
       messageBox("Device time out",
        mfError | mfOKButton);
       return 0;
     }

   if (status & 0x20)
     {
       puts("\a");
       messageBox("Out of paper",
        mfError | mfOKButton);
       return 0;
     }

   if (status & 0x08)
     {
       puts("\a");
       messageBox("I/O error.",
        mfError | mfOKButton);
       return 0;
     }

   if (status & 0x10)
     {
       suspend();
       gotoxy(1,10);

       if((fpout=fopen("lpt1","w"))==NULL)
         {
          puts("\a");
          messageBox("No se puede imprimir",
           mfError | mfOKButton);
          return 0;
         }


       _setcursortype(_NOCURSOR);
       puts("I M P R I M I E N D O                    \n");
```

```
        textattr(207);
        cprinti("Espere un momento");

        fprintf(fpout,"\t\tPROGRAMA SEMIGRAFICO PARA ANALIZAR ESTABLIDAD");
        fprintf(fpou.,"\n\t\t        USANDO LA FUNCION DESCRIPTIVA");


        fprintf(fpout,"\n\n\t\tFUNCION DE TRANSFERENCIA");
        fprintf(fpout,"\n\n\tGrado del Numerador-   %d",gradnum);
        fprintf(fpout,"\n\tGrado del Denominador- %d",gradden);

        fprintf(fpout,"\n\n\tCOEFICIENTES");
        fprintf(fpout,"\n\n\tNUMERADOR");
        for(i=0;i<=gradnum;i++)
          fprintf(fpout,"\n\ta%d- %f",i,numerador[i]);
        fprintf(fpout,"\n\n\tDENOMINADOR");
        for(i=0;i<=gradden;i++)
          fprintf(fpout,"\n\tb%d- %f",i,denominador[i]);

        fprintf(fpout,"\n\n\t\tRESPUESTA DE FRECUENCIA DE LA FUNCION DE TRANSFERENCIA\n");
        inicio=factor_dato/(float)longit_vect;
        fprintf(fpout, "\t   %s\t%s\t%s\t%s\t%s\n",
                 "W", "Real FT.", "Imag. FT.", "Módulo FT", "Angulo FT");
        for(i=0; i<longit_vect; i++)
          {
           status = biosprint(_PRINTER_STATUS, 0, 0);
           if (status & 0x08)
             {
               puts("\a");
               resume();
               redraw();
               return 0;
             }
           if(i<80)
             w=inicio*(i+1);
           if(i>=80 && i<=190)
             w=0.4*i-27.6;
           if(i>190)
             w=2*i-331.6;

           Mod = sqrt((double)realFT[i]*realFT[i]+imagFT[i]*imagFT[i]);
           if(realFT[i]!=0)
             {
               Ang = atan((double)imagFT[i]/realFT[i])*180/pi;
               if(realFT[i]<0)
                 Ang = 180 + Ang;
               if(realFT[i]>0 && imagFT[i]<0)
                 Ang = 360 + Ang;
             }
           else
             {
               if(imagFT[i]<0)
                 Ang = 270;
               else
                 Ang = 90;
             }

           fprintf(fpout,"\t%6.3f\t%9.3e\t%9.3e\t%9.3e\t%6.3f\n",
                     w, realFT[i], imagFT[i], Mod, Ang);
          }
        fclose(fpout);
       }
     resume();
     redraw();
    }
  return 0;
}
```

```
/* FUNCION PARA IMPRIMIR LOS RESULTADOS DE LA FUNCION DE TRANSFERENCIA COMPENSADA*/
int TVAesfudes::imprimirGc()
{
    int i, status;
    float dato, w, inicio, Mod, Ang;
    FILE *fpout;

    i=0;
    TDialog *imprimir = new TDialog(TRect(0, 0, 40, 12), "Imprimir");

    imprimir->insert(
      new TStaticText(TRect(5, 2, 35, 8),
        "\003IMPRESION DE LA RESPUESTA DE FRECUENCIA\n"
        "\003DE LA PLANTA CON COMPENSACION\n\003\n"
        "\003Asegurese de que\n"
        "\003La Impresora este Lista?"
        )
      );

    imprimir->insert(
      new TButton(TRect(7, 9, 17, 11), "~O~K", cmOK, bfNormal)
      );
    imprimir->insert(
      new TButton(TRect(23, 9, 33, 11), "~C~ancel", cmCancel, bfDefault)
      );
    imprimir->options |= ofCentered;

    ushort control=deskTop->execView(imprimir);
    if(control == cmOK)
      i=1;

    destroy( imprimir);

    if (i==1)
      {
       status = biosprint(_PRINTER_STATUS, 0, 0);
       if (status & 0x01)
         {
           puts("\a");
           messageBox("Device time out",
            mfError | mfOKButton);
           return 0;
         }

       if (status & 0x20)
         {
           puts("\a");
           messageBox("Out of paper",
            mfError | mfOKButton);
           return 0;
         }

       if (status & 0x08)
         {
           puts("\a");
           messageBox("I/O error.",
            mfError | mfOKButton);
           return 0;
         }

       if (status & 0x10)
         {
           suspend();
           gotoxy(1,10);

           if((fpout=fopen("lpt1","w"))==NULL)
```

```
    puts("\a");
    messageBox("No se puede imprimir",
     mfError | mfOKButton);
    return 0;
   }


_setcursortype(_NOCURSOR);
puts("I M P R I M I E N D O .............\n");
textattr(207);
cprintf("Espere un momento");

fprintf(fpout,"\t\tPROGRAMA SEMIGRAFICO PARA ANALIZAR ESTABLIDAD");
fprintf(fpout,"\n\t\t        USANDO LA FUNCION DESCRIPTIVA");


fprintf(fpout,"\n\n\t\tFUNCION DE TRANSFERENCIA");
fprintf(fpout,"\n\n\tGrado del Numerador=   %d",gradnum);
fprintf(fpout,"\n\tGrado del Denominador= %d",gradden);

fprintf(fpout,"\n\n\tCOEFICIENTES");
fprintf(fpout,"\n\n\tNUMERADOR");
for(i=0;i<=gradnum;i++)
  fprintf(fpout,"\n\ta%d= %f",i,numerador[i]);
fprintf(fpout,"\n\n\tDENOMINADOR");
for(i=0;i<=gradden;i++)
  fprintf(fpout,"\n\tb%d= %f",i,denominador[i]);


fprintf(fpout,"\n\n\t\tCOMPENSACION PID");
fprintf(fpout,"\n\n\tKp =   %f",Kp);
fprintf(fpout,"\n\tKd = %f",Kd);
fprintf(fpout,"\n\tKi = %f",Ki);

fprintf(fpout,"\n\n\t\tRESPUESTA DE FRECUENCIA DE LA FT COMPENSADA\n");
inicio=factor_dato/(float)longit_vect;
fprintf(fpout, "\t   %s\t%s\t%s\t%s\t%s\n",
         "W", "Real FTc.", "Imag. FTc.", "Módulo FTc", "Angulo FTc");
for(i=0; i<longit_vect; i++)
  {
   status = biosprint(_PRINTER_STATUS, 0, 0);
   if (status & 0x08)
     {
       puts("\a");
       resume();
       redraw();
       return 0;
     }
   if(i<80)
     w=inicio*(i+1);
   if(i>=80 && i<=190)
     w=0.4*i-27.6;
   if(i>190)
     w=2*i-331.6;

   Mod = sqrt((double)realFTc[i]*realFTc[i]+imagFTc[i]*imagFTc[i]);
   if(realFTc[i]!=0)
     {
       Ang = atan((double)imagFTc[i]/realFTc[i])*180/pi;
       if(realFTc[i]<0)
         Ang = 180 + Ang;
       if(realFTc[i]>0 && imagFTc[i]<0)
         Ang = 360 + Ang;
     }
   else
     {
       if(imagFTc[i]<0)
         Ang = 270;
       else
```

```
                    Ang = 90;
                  }

                fprintf(fpout,"\t%6.3f\t%9.3e\t%9.3e\t%9.3e\t%6.3f\n",
                        W, realFTc[i], imagFTc[i], Mod, Ang);
              }
            fclose(fpout);
          }
        resume();
        redraw();
      }
    return 0;
}



/* FUNCION PARA IMPRIMIR LOS RESULTADOS DE LA FUNCION DESCRIPTIVA*/
int TVAesfudes::imprimirFD()
  {
    int i, status;
    float dato, W, inicio, A, Mod, Ang;
    FILE *fpout;

    i=0;
    TDialog *imprimir = new TDialog(TRect(0, 0, 35, 12), "Imprimir");

    imprimir->insert(
      new TStaticText(TRect(5, 2, 30, 8),
        "\003IMPRESION DE LA RESPUESTA\n"
        "\003FRECUENCIA DE LA ALINEALIDAD\n\003\n"
        "\003Asegurese de que\n"
        "\003La Impresora este Lista?"
        )
      );

    imprimir->insert(
      new TButton(TRect(5, 9, 15, 11), "~O~K", cmOK, bfNormal)
      );
    imprimir->insert(
      new TButton(TRect(20, 9, 30, 11), "~C~ancel", cmCancel, bfDefault)
      );
    imprimir->options |= ofCentered;

    ushort control=deskTop->execView(imprimir);
    if(control == cmOK)
      i=1;

    destroy( imprimir);


    if (i==1)
      {
        status = biosprint(_PRINTER_STATUS, 0, 0);
        if (status & 0x01)
          {
            puts("\a");
            messageBox("Device time out",
             mfError | mfOKButton);
            return 0;
          }

        if (status & 0x20)
          {
            puts("\a");
            messageBox("Out of paper",
             mfError | mfOKButton);
```

```
  }

if (status & 0x08)
  {
    puts("\a");
    messageBox("I/O error.",
     mfError | mfOKButton);
    return 0;
  }

if (status & 0x10)
  {
    suspend();
    gotoxy(1,10);
    biosprint(_PRINTER_INIT, 0, 0);

    if((fpout=fopen("lpt1","w"))==NULL)
      {
       puts("\a");
       messageBox("No se puede imprimir",
        mfError | mfOKButton);
       return 0;
      }

    _setcursortype(_NOCURSOR);
    puts("I M P R I M I E N D O .............\n");
    textattr(207);
    cprintf("Espere un momento");

    fprintf(fpout,"\n\n\t\tPROGRAMA SEMIGRAFICO PARA ANALIZAR ESTABLIDAD");
    fprintf(fpout,"\n\t\t        USANDO LA FUNCION DESCRIPTIVA");

    fprintf(fpout,"\n\n\t\tFUNCION DESCRIPTIVA\n");
    if(tipoNL==0)
      fprintf(fpout,"\t\tNO HAY ALINEALIDAD ESCOGIDA");
    if(tipoNL==1)
      {
       fprintf(fpout,"\tRELE IDEAL\n");
       fprintf(fpout,"\tU = %f\n",pto1);
      }
    if(tipoNL==2)
      {
       fprintf(fpout,"\tRELE CON ZONA MUERTA\n");
       fprintf(fpout,"\tU = %f\n",pto1);
       fprintf(fpout,"\tD = %f\n",pto2);
      }
    if(tipoNL==3)
      {
       fprintf(fpout,"\tRELE CON PENDIENTE\n");
       fprintf(fpout,"\tU = %f\n",pto1);
       fprintf(fpout,"\tK = %f\n",pto2);
      }
    if(tipoNL==4)
      {
       fprintf(fpout,"\tRELE CON ZONA MUERTA Y PENDIENTE\n");
       fprintf(fpout,"\tU = %f\n",pto1);
       fprintf(fpout,"\tD = %f\n",pto2);
       fprintf(fpout,"\tK = %f\n",pto3);
      }
    if(tipoNL==5)
      {
       fprintf(fpout,"\tSATURACION IDEAL\n");
       fprintf(fpout,"\tU = %f\n",pto1);
       fprintf(fpout,"\tD = %f\n",pto2);
      }
    if(tipoNL==6)
      {
       fprintf(fpout,"\tSATURACION CON ZONA MUERTA\n");
       fprintf(fpout,"\tU = %f\n",pto1);
```

```
    fprintf(fpout,"\tD1- %f\n",pto2);
    fprintf(fpout,"\tD2- %f\n",pto3);
  }
if(tipoNL--7)
  {
    fprintf(fpout,"\tSATURACION CON PENDIETE\n");
    fprintf(fpout,"\tU - %f\n",pto1);
    fprintf(fpout,"\tD - %f\n",pto2);
    fprintf(fpout,"\tK - %f\n",pto3);
  }
if(tipoNL==8)
  {
    fprintf(fpout,"\tSATURACION CON ZONA MUERTA Y PENDIENTE\n");
    fprintf(fpout,"\tU = %f\n",pto1);
    fprintf(fpout,"\tD1- %f\n",pto2);
    fprintf(fpout,"\tD2- %f\n",pto3);
    fprintf(fpout,"\tK - %f\n",pto4);
  }
if(tipoNL--9)
  {
    fprintf(fpout,"\tHISTERESIS IDEAL\n");
    fprintf(fpout,"\tU - %f\n",pto1);
    fprintf(fpout,"\tD - %f\n",pto2);
  }
if(tipoNL--10)
  {
    fprintf(fpout,"\tHISTERESIS CON ZONA MUERTA\n");
    fprintf(fpout,"\tU - %f\n",pto1);
    fprintf(fpout,"\tD1- %f\n",pto2);
    fprintf(fpout,"\tD2- %f\n",pto3);
  }
if(tipoNL--11)
  {
    fprintf(fpout,"\tHISTERESIS CON PENDIENTE\n");
    fprintf(fpout,"\tU - %f\n",pto1);
    fprintf(fpout,"\tD1- %f\n",pto2);
    fprintf(fpout,"\tD2- %f\n",pto3);
  }
if(tipoNL--12)
  {
    fprintf(fpout,"\tHISTERESIS CON ZONA MUERTA Y PENDIENTE\n");
    fprintf(fpout,"\tU - %f\n",pto1);
    fprintf(fpout,"\tD1- %f\n",pto2);
    fprintf(fpout,"\tD2- %f\n",pto3);
    fprintf(fpout,"\tD3- %f\n",pto4);
  }


if(tipoNL!-0)
  {
    fprintf(fpout,"\n\n\t\tRESPUESTA DE FRECUENCIA DE LA FUNCION DESCRIPTIVA\n");
    fprintf(fpout, "\t   %s\t%s\t%s\t%s\t%s\n",
             "A", "Real FD.", "Imag. FD.", "Módulo FD", "Angulo FD");
    for(i-0; i<num_datosFD; i++)
      {
        status - biosprint(_PRINTER_STATUS, 0, 0);
        if (status & 0x08)
          {
            puts("\a");
            resume();
            redraw();
            return 0;
          }

        if(i<-20)
          A-0.3*i+inicioAFD;
        else
          A-2*i-34+inicioAFD;
```

```
                    Mod = sqrt((double)realFD[i]*realFD[i]+imagFD[i]*imagFD[i]);
                    if(realFD[i]!=0)
                      {
                       Ang = atan((double)imagFD[i]/realFD[i])*180/pi;
                       if(realFD[i]<0)
                         Ang = 180 + Ang;
                       if(realFD[i]>0 && imagFD[i]<0)
                         Ang = 360 + Ang;
                      }
                    else
                      {
                       if(imagFD[i]<0)
                         Ang = 270;
                       else
                         Ang = 90;
                      }

                    fprintf(fpout,"  \t%6.2f\t%9.3e\t%9.3e\t%9.3e\t%6.3f\n",
                             A, realFD[i], imagFD[i], Mod, Ang);
                  }
              ]

            fclose(fpout);
          }
        resume();
        redraw();
      ]
    return 0;
  ]


int TVAesfudes::imprimirTime()
{
  int i, status;
  FILE *fpout;

  i=0;
  TDialog *imprimir = new TDialog(TRect(0, 0, 40, 12), "Imprimir");

  imprimir->insert(
    new TStaticText(TRect(5, 2, 35, 8),
      "\003IMPRESION DE LA RESPUESTA\n"
      "\003DEL SISTEMA EN EL TIEMPO\n\003\n"
      "\003Asegurese de que\n"
      "\003La Impresora este Lista?"
      )
    );

  imprimir->insert(
    new TButton(TRect(7, 9, 17, 11), "~O~K", cmOK, bfNormal)
    );
  imprimir->insert(
    new TButton(TRect(23, 9, 33, 11), "~C~ancel", cmCancel, bfDefault)
    );
  imprimir->options |= ofCentered;

  ushort control=deskTop->execView(imprimir);
  if(control == cmOK)
    i=1;

  destroy( imprimir);

  if (i==1)
    [
      status = biosprint(_PRINTER_STATUS, 0, 0);
      if (status & 0x01)
       {
         puts("\a");
```

```
    messageBox("Device time out",
     mfError | mfOKButton);
    return 0;
  }

if (status & 0x20)
  {
    puts("\a");
    messageBox("Out of paper",
     mfError | mfOkButton);
    return 0;
  }

if (status & 0x08)
  {
    puts("\a");
    messageBox("I/O error.",
     mfError | mfOKButton);
    return 0;
  }

if (status & 0x10)
  {
    suspend();
    gotoxy(1,10);

    if((fpout=fopen("lpt1","w"))==NULL)
      {
        puts("\a");
        messageBox("No se puede imprimir",
         mfError | mfOKButton);
        return 0;
      }


    _setcursortype(_NOCURSOR);
    puts("I M P R I M I E N D O .............\n");
    textattr(207);
    cprintf("Espere un momento");

    fprintf(fpout,"\t\tPROGRAMA SEMIGRAFICO PARA ANALIZAR ESTABLIDAD");
    fprintf(fpout,"\n\t\t        USANDO LA FUNCION DESCRIPTIVA");


    fprintf(fpout,"\n\n\t\tFUNCION DE TRANSFERENCIA");
    fprintf(fpout,"\n\n\tGrado del Numerador=   %d",gradnum);
    fprintf(fpout,"\n\tGrado del Denominador= %d",gradden);

    fprintf(fpout,"\n\n\tCOEFICIENTES");
    fprintf(fpout,"\n\n\tNUMERADOR");
    for(i=0;i<=gradnum;i++)
      fprintf(fpout,"\n\ta%d= %f",i,numerador[i]);
    fprintf(fpout,"\n\n\tDENOMINADOR");
    for(i=0;i<=gradden;i++)
      fprintf(fpout,"\n\tb%d= %f",i,denominador[i]);

    if(hayGc!=0)
      {
        fprintf(fpout,"\n\n\t\tCOMPENSACION PID");
        fprintf(fpout,"\n\n\tKp =   %f",Kp);
        fprintf(fpout,"\n\tKd = %f",Kd);
        fprintf(fpout,"\n\tKi = %f",Ki);
      }

    if(tipoNL!=0)
      {
        fprintf(fpout,"\n\n\t\tFUNCION DESCRIPTIVA\n");
        if(tipoNL==1)
```

```
    fprintf(fpout,"\tRELE IDEAL\n");
    fprintf(fpout,"\tU - %f\n",pto1);
  }
 if(tipoNL--2)
  {
    fprintf(fpout,"\tRELE CON ZONA MUERTA\n");
    fprintf(fpout,"\tU - %f\n",pto1);
    fprintf(fpout,"\tD - %f\n",pto2);
  }
 if(tipoNL--3)
  {
    fprintf(fpout,"\tRELE CON PENDIENTE\n");
    fprintf(fpout,"\tU - %f\n",pto1);
    fprintf(fpout,"\tK - %f\n",pto2);
  }
 if(tipoNL--4)
  {
    fprintf(fpout,"\tRELE CON ZONA MUERTA Y PENDIENTE\n");
    fprintf(fpout,"\tU - %f\n",pto1);
    fprintf(fpout,"\tD - %f\n",pto2);
    fprintf(fpout,"\tK - %f\n",pto3);
  }
 if(tipoNL--5)
  {
    fprintf(fpout,"\tSATURACION IDEAL\n");
    fprintf(fpout,"\tU - %f\n",pto1);
    fprintf(fpout,"\tD = %f\n",pto2);
  }
 if(tipoNL--6)
  {
    fprintf(fpout,"\tSATURACION CON ZONA MUERTA\n");
    fprintf(fpout,"\tU - %f\n",pto1);
    fprintf(fpout,"\tD1- %f\n",pto2);
    fprintf(fpout,"\tD2- %f\n",pto3);
  }
 if(tipoNL--7)
  {
    fprintf(fpout,"\tSATURACION CON PENDIETE\n");
    fprintf(fpout,"\tU - %f\n",pto1);
    fprintf(fpout,"\tD - %f\n",pto2);
    fprintf(fpout,"\tK - %f\n",pto3);
  }
 if(tipoNL--8)
  {
    fprintf(fpout,"\tSATURACION CON ZONA MUERTA Y PENDIENTE\n");
    fprintf(fpout,"\tU = %f\n",pto1);
    fprintf(fpout,"\tD1- %f\n",pto2);
    fprintf(fpout,"\tD2- %f\n",pto3);
    fprintf(fpout,"\tK - %f\n",pto4);
  }
 if(tipoNL--9)
  {
    fprintf(fpout,"\tHISTERESIS IDEAL\n");
    fprintf(fpout,"\tU - %f\n",pto1);
    fprintf(fpout,"\tD - %f\n",pto2);
  }
 if(tipoNL--10)
  {
    fprintf(fpout,"\tHISTERESIS CON ZONA MUERTA\n");
    fprintf(fpout,"\tU - %f\n",pto1);
    fprintf(fpout,"\tD1- %f\n",pto2);
    fprintf(fpout,"\tD2- %f\n",pto3);
  }
 if(tipoNL--11)
  {
    fprintf(fpout,"\tHISTERESIS CON PENDIENTE\n");
    fprintf(fpout,"\tU - %f\n",pto1);
    fprintf(fpout,"\tD1- %f\n",pto2);
    fprintf(fpout,"\tD2- %f\n",pto3);
```

```
            }
          if(tipoNL==12)
            {
              fprintf(fpout,"\tHISTERESIS CON ZONA MUERTA Y PENDIENTE\n");
              fprintf(fpout,"\tU - %f\n",pto1);
              fprintf(fpout,"\tD1- %f\n",pto2);
              fprintf(fpout,"\tD2- %f\n",pto3);
              fprintf(fpout,"\tD3- %f\n",pto4);
            }
        }

      fprintf(fpout,"\n\n\t\tRESPUESTA EN EL TIEMPO\n");
      fprintf(fpout, "\t   %s\t%s\n", "t[s]", "Yk");
      for(i=0; i<800; i++)
        {
          status = biosprint(_PRINTER_STATUS, 0, 0);
          if (status & 0x08)
            {
              puts("\a");
              resume();
              redraw();
              return 0;
            }
          fprintf(fpout,"\t%6.4f\t%9.3e\n", T*25*i, Yk[i]);
        }
      fclose(fpout);
    }
  resume();
  redraw();
 }
return 0;
}
```

```
/*-------------------------------------------------------*/
/*                                                       */
/*   PROGRAMA PARA ANALIZAR ESTABILIDAD                  */
/*   UTILIZANDO LA FUNCION DESCRIPTIVA                   */
/*                                                       */
/*      FUNCION DE TRANSFERENCIA                         */
/*                                                       */
/*   Wilmer Cárdenas Burbano 1994                        */
/*-------------------------------------------------------*/

#define Uses_TApplication
#define Uses_TDialog
#define Uses_MsgBox
#define Uses_TDeskTop
#define Uses_TStaticText
#define Uses_TButton
#define Uses_TInputLine

#include <tv.h>
#include <stdio.h>
#include <math.h>
#include <conio.h>
#include <ctype.h>
#include <graphics.h>
#include <string.h>

#include "funcion.h"

extern int gradnum, gradden;
extern float numerador[];
extern float denominador[];
extern float realFT[];
extern float imagFT[];
extern float realFTc[];
extern float imagFTc[];

extern int longit_vect;
extern float factor_dato;
extern int hayFT, hayGc;
extern float Kp, Kd, KI;


int TVAesfudes::datosnuevos()
{
    int i, j;
    int xmax, ymax;
    char ch;
    gradnum=0;
    gradden=0;
    const MAXSIZE = 80;
    char errorMsg[MAXSIZE];

    suspend();
    j=1;
    if (graphicsStart() == False)
      {
       strcpy(errorMsg,grapherrormsg(graphresult()));
       strcat( errorMsg,"." );
       messageBox(errorMsg, mfError | mfOKButton);
       j=0;
      }
    else
      {
       setbkcolor(BLUE);
       /*INICIALIZACION DE ARREGLOS*/
       for(i=0;i<12;i++)
         numerador[i]=0;
       for(i=0;i<12;i++)
         denominador[i]=0;
```

```
        for(i=0;i<200;i++)
          realFT[i]=0;
        for(i=0;i<200;i++)
          imagFT[i]=0;
        xmax = getmaxx();
        ymax = getmaxy();
        settextjustify(0,0);
        do
          {
            i=1;
            cleardevice();
            settextjustify(1,1);
            setcolor(WHITE);
              outtextxy(xmax/2,5,"PROGRAMA SEMIGRAFICO PARA ANALIZAR ESTABILIDAD USANDO LA
FUNCION DESCRIPTIVA");
            moveto(0,0);
            setcolor(GREEN);
            gotoxy(1,2);
            puts("Ingrese la FUNCION DE TRANSFERENCIA de la parte lineal\n");
            settextjustify(0,0);
            outtextxy(160,70,"  a0+a1s +.....+ans");
            outtextxy(160,90," b0+b1s +.....+bms");
            outtextxy(160,67,"        1              n");
            outtextxy(160,87,"       1               m");
            outtextxy(120,79,"FT= ");
            setlinestyle(0,0,1);
            line(150,75,400,75);
            gotoxy(5,8);
            setcolor(YELLOW);
            puts("Ingrese el grado del Numerador");
            do
              {
               gotoxy(1,10);
               puts("                                          ");
               gotoxy(45,8);
               puts("                  ");
               puts("                                            \n");
               puts("                                           ");
               gotoxy(45,8);
               while(scanf("%d",&gradnum)--0)
                 {
                   fflush(stdin);
                   gotoxy(45,8);
                   printf("\a       ");
                   gotoxy(1,9);
                   puts("                                ");
                   gotoxy(45,8);
                 }
               gotoxy(45,8);
               printf("%d",gradnum);
               gotoxy(1,9);
               puts("                                ");

               gotoxy(5,9);
               puts("Ingrese el grado del denominador");
               gotoxy(45,9);
               while(scanf("%d",&gradden)--0)
                 {
                   fflush(stdin);
                   gotoxy(45,9);
                   printf("\a       ");
                   gotoxy(1,10);
                   puts("                                          ");
                   gotoxy(45,9);
                 }
               gotoxy(45,9);
               printf("%d",gradden);

               setcolor(RED);
```

```
            printf("\nLOS DATOS SON CORRECTOS s/n\t");
            setcolor(YELLOW);
            fflush(stdin);
            ch=getch();
            if(isupper(ch))
              ch=tolower(ch);
            if(ch!='s')
              puts("\a");
            if(gradnum>11||gradden>11)
              {
                gotoxy(1,12);
                setcolor(LIGHTRED);
                puts("\aEl máximo grado permitido es 11");
                setcolor(YELLOW);
                ch='n';
              }
          }while(ch!='s');
        gotoxy(1,10);
        puts("                                          ");

        /*VER SI ES UN SISTEMA FISICAMENTE REALIZABLE*/
        if(gradnum>=gradden)
          {
            printf("\a");
            settextjustify(1,1);
            moveto(xmax/2, ymax/2);
            setcolor(LIGHTRED);
            outtext("ERROR NO ES UN SISTEMA FISICAMENTE REALIZABLE");
            gotoxy(39,25);
            setcolor(MAGENTA);
            puts("Presione una tecla e intente de nuevo");
            fflush(stdin);
            getch();
            i=0;
          }
      }while(i--0);
  }
  return j;
}



int TVAesfudes::datosgrabados()
{
  char archivo[30];
  float dato;
  FILE *fpin;
  int i, j, k;

  for(i=0;i<30;i++)
    archivo[i]='\0';

  i=0;
  TDialog *abrir = new TDialog(TRect(0, 0, 40, 13), "Abrir una FT");

  abrir->insert(
    new TStaticText(TRect(5, 2, 35, 8),
      "\003UTILIZAR UNA FT GRABADA\n"
      "\003ANTERIORMENTE\n\003\n"
      "\003Ingrese el nombre\n"
      "\003del archivo de entrada\n"
      "\003(sin extensión)"
      )
    );

  abrir->insert(new TInputLine(TRect(10,8,30,9),30));
```

```
abrir->insert(
  new TButton(TRect(7, 10, 17, 12), "~O~K", cmOK, bfDefault)
  );
abrir->insert(
  new TButton(TRect(23, 10, 33, 12), "~C~ancel", cmCancel, bfNormal)
  );
abrir->options |= ofCentered;

abrir->setData(archivo);

ushort control=deskTop->execView(abrir);
k=0;
if(control == cmOK)
  {
   i=1;
   k=1;
   abrir->getData(archivo);
  }
destroy(abrir);


if(i==1)
  {
   /*INICIALIZACION DE ARREGLOS*/
   gradnum = 0;
   gradden = 0;

   for(i=0;i<12;i++)
     numerador[i]=0;
   for(i=0;i<12;i++)
     denominador[i]=0;
   for(i=0;i<200;i++)
     realFT[i]=0;
   for(i=0;i<200;i++)
     imagFT[i]=0;

   for(i=1;i<30;i++)
     {
       if(archivo[i]=='\0')
         {
          j=i;
          i=30;
         }
     }

   archivo[j]='.';
   archivo[j+1]='c';
   archivo[j+2]='n';
   archivo[j+3]='l';

   if((fpin=fopen(archivo,"r"))==NULL)
     {
       puts("\a");
       messageBox("No se puede leer el archivo",
        mfError | mfOKButton);
       return 0;
     }

   fscanf(fpin,"%d",&gradnum);
   fscanf(fpin,"%d",&gradden);

   if(gradnum>=gradden)
     {
       puts("\a");
       messageBox("NO ES UN SISTEMA FISICAMENTE REALIZABLE",
        mfError | mfOKButton);
       gradnum = 0;
       gradden = 0;
       hayFT=0;
```

```
            return 0;
          }

        for(i=0;i<=gradnum;i++)
          {
            fscanf(fpin,"%f",&dato);
            numerador[i]=dato;
          }
        for(i=0;i<=gradden;i++)
          {
            fscanf(fpin,"%f",&dato);
            denominador[i]=dato;
          }
      }
  return k;
}



/*FUNCION DE INGRESO DE DATOS DEL NUMERADOR*/
void TVAesfudes::i_numerador()
  {
    int i, Y;
    float dato;
    char ch;

    gotoxy(1,11);
    puts("                                ");
    gotoxy(1,12);
    puts("                                          ");
    gotoxy(10,11);
    puts("NUMERADOR");
    do
      {
        for(i=0;i<12;i++)
          numerador[i]=0;
        gotoxy(10,12);
        puts("                                                  ");
        puts("                                                  ");
        puts("                                                  ");
        puts("                                                  ");
        puts("                                                  ");
        puts("                                                  ");
        puts("                                                  ");
        puts("                                                  ");
        puts("                                                  ");
        puts("                                                  ");
        puts("                                                  ");
        puts("                                                  ");

        Y=12;
        for(i=0;i<=gradnum;i++)
          {
            gotoxy(10,Y+i);
            printf("a%d= ",i);
            fflush(stdin);
            gotoxy(14,Y+i);
            while(scanf("%f",&dato)==0)
              {
                fflush(stdin);
                gotoxy(14,Y+i);
                printf("\a       ");
                gotoxy(1,Y+i+1);
                puts("                                    ");
                gotoxy(14,Y+i);
              }
            gotoxy(1,Y+i+1);
            puts("                                    ");
```

```
            gotoxy(14,Y+1);
            printf("%f",dato);
            numerador[i]-dato;
          }
        gotoxy(10,Y+gradnum+1);
        setcolor(RED);
        puts("LOS DATOS SON CORRECTOS s/n\t");
        setcolor(YELLOW);
        fflush(stdin);
        ch-getch();
        if(isupper(ch))
        ch-tolower(ch);

        if(ch!='s')
        puts("\a");

      }while(ch!-'s');
    gotoxy(10,Y+gradnum+1);
    puts("                              ");
  }



/*FUNCION DE INGRESO DE DATOS DEL DENOMINADOR*/
void TVAesfudes::i_denominador()
  {
    int i, Y;
    float dato;
    char ch;

    gotoxy(40,11);
    printf("DENOMINADOR");
    do
      {
        for(i-0;i<12;i++)
          denominador[i]-0;
        gotoxy(40,12);
        puts("                              ");
        gotoxy(40,13);
        puts("                              ");
        gotoxy(40,14);
        puts("                              ");
        gotoxy(40,15);
        puts("                              ");
        gotoxy(40,16);
        puts("                              ");
        gotoxy(40,17);
        puts("                              ");
        gotoxy(40,18);
        puts("                              ");
        gotoxy(40,19);
        puts("                              ");
        gotoxy(40,20);
        puts("                              ");
        gotoxy(40,21);
        puts("                              ");
        gotoxy(40,12);
        puts("                              ");
        gotoxy(40,23);
        puts("                              ");
        gotoxy(40,24);
        puts("                              ");

        Y-12;
        for(i-0;i<-gradden;i++)
          {
            gotoxy(40,Y+i);
            printf("b%d- ",i);
```

```
            fflush(stdin);
            gotoxy(44,Y+i);
            while(scanf("%f",&dato)--0)
              {
               fflush(stdin);
               gotoxy(44,Y+i);
               printf("\a       ");
               gotoxy(1,Y+i+1);
               puts("         ");
               gotoxy(44,Y+i);
              }
            gotoxy(1,Y+i+1);
            puts("       . ");
            gotoxy(44,Y+i);
            printf("%f",dato);
            denominador[i]-dato;
          }
        gotoxy(40,Y+gradden+1);
        setcolor(RED);
        puts("LOS DATOS SON CORRECTOS s/n\t");
        setcolor(YELLOW);
        fflush(stdin);
        ch-getch();
        if(isupper(ch))
        ch-tolower(ch);

        if(ch!='s')
        puts("\a");

      }while(ch!='s');

    graphicsStop();
    resume();

  }



 /*FUNCION DE CALCULO DE LAS COORDENADAS DE LA FT*/
void TVAesfudes::coordenadasFT()
{
    int i, j;
    float  w, inicio, denom;
    float realnumerad, imagnumerad, realdenom, imagdenom;
    inicio=factor_dato/(float)longit_vect;
    hayGc=0;

    for(j-0;j<longit_vect;j++)
      {
       if(j<80)
         w-inicio*(j+1);
       if(j>-80 && j<-190)
         w=0.4*j-27.6;
       if(j>190)
         w-2*j-331.6;

       realnumerad-0;
       for(i=0;i<=gradnum;i+=2)
          realnumerad-realnumerad+pow(-1,i/2)*numerador[i]*pow(w,i);

       imagnumerad=0;
       for(i-1;i<-gradnum;i+-2)
          imagnumerad-imagnumerad+pow(-1,(i-1)/2)*numerador[i]*pow(w,i);

       realdenom-0;
       for(i=0;i<=gradden;i+=2)
          realdenom-realdenom+pow(-1,i/2)*denominador[i]*pow(w,i);
```

```
        imagdenom=0;
        for(i=1;i<=gradden;i+=2)
           imagdenom=imagdenom+pow(-1,(i-1)/2)*denominador[i]*pow(w,i);

        denom = pow(realdenom,2)+pow(imagdenom,2);

        if (denom!=0)
          {
            hayFT=1;
            realFT[j] =(realnumerad*realdenom+imagnumerad*imagdenom)/denom;
            imagFT[j] =(imagnumerad*realdenom-realnumerad*imagdenom)/denom;
          }
        else
          {
            j=longit_vect;
          }
      }

    if (denom==0)
      {
        puts("\a");
        messageBox("LOS DATOS SON INCONSISTENTES",
         mfError | mfOKButton);
        for(i=0;i<200;i++)
          realFT[i]=0;
        for(i=0;i<200;i++)
          imagFT[i]=0;
      }

    if (denom!=0 && denominador[gradden]==0)
      {
        puts("\a");
        messageBox("LOS DATOS SON INCONSISTENTES, bm no puede ser igual a cero",
         mfError | mfOKButton);
        for(i=0;i<200;i++)
          realFT[i]=0;
        for(i=0;i<200;i++)
          imagFT[i]=0;
        hayFT=0;
      }
}



//COMPENSACION

void TVAesfudes::IngresoGc()
{
    int i, j, xmax;
    float  w, inicio;
    char ch;

    suspend();
    graphicsStart();
    setbkcolor(BLUE);

    /*INICIALIZACION DE ARREGLOS*/
    for(i=0;i<200;i++)
      realFTc[i]=0;
    for(i=0;i<200;i++)
      imagFTc[i]=0;

    xmax = getmaxx();
    settextjustify(0,0);

    cleardevice();
    settextjustify(1,1);
    setcolor(WHITE);
```

```
    outtextxy(xmax/2,5,"PROGRAMA SEMIGRAFICO PARA ANALIZAR ESTABILIDAD USANDO LA FUNCION
DESCRIPTIVA");
    moveto(0,0);
    setcolor(GREEN);
    gotoxy(1,2);
    puts("Ingrese la COMPENSACION TIPO PID DESEADA\n");
    settextjustify(0,0);
    outtextxy(160,70,"Gc = Kp(1+ Ki/s + Kd*s)");
    setcolor(YELLOW);

    do
      {
       do
         {
          gotoxy(1,10);
          puts("                                    ");
          gotoxy(1,12);
          puts("                                    ");
          gotoxy(1,14);
          puts("                                    ");
          gotoxy(1,18);
          puts("                                    ");

          gotoxy(10,10);
          puts("Kp = ");
          gotoxy(15,10);
          while(scanf("%f",&Kp)==0)
            {
             fflush(stdin);
             printf("\a       ");
             gotoxy(15,10);
             puts("                                 ");
             gotoxy(1,11);
             puts("                                 ");
             gotoxy(1,12);
             puts("                                 ");
             gotoxy(15,10);
            }
          gotoxy(15,10);
          printf("%f",Kp);
          gotoxy(1,11);
          puts("                                    ");
          gotoxy(1,12);
          puts("                                    ");

          gotoxy(10,12);
          puts("Kd = ");
          gotoxy(15,12);
          while(scanf("%f",&Kd)--0)
            {
             fflush(stdin);
             printf("\a       ");
             gotoxy(15,12);
             puts("                                 ");
             gotoxy(1,13);
             puts("                                 ");
             gotoxy(1,14);
             puts("                                 ");
             gotoxy(15,12);
            }
          gotoxy(15,12);
          printf("%f",Kd);
          gotoxy(1,13);
          puts("                                    ");
          gotoxy(1,14);
          puts("                                    ");


          gotoxy(10,14);
```

```
      puts("Ki - ");
      gotoxy(15,14);
      while(scanf("%f",&Ki)==0)
        {
         fflush(stdin);
         printf("\a       ");
         gotoxy(15,14);
         puts("                                  ");
         gotoxy(1,15);
         puts("                                  ");
         gotoxy(1,16);
         puts("                                  ");
         gotoxy(15,14);
        }
      gotoxy(15,14);
      printf("%f",Ki);
      gotoxy(1,15);
      puts("                                  ");
      gotoxy(1,16);
      puts("                                  ");

      setcolor(RED);
      printf("\nLOS DATOS SON CORRECTOS s/n\t");
      setcolor(YELLOW);
      fflush(stdin);
      ch=getch();
      if(isupper(ch))
        ch=tolower(ch);
      if(ch!='s')
        puts("\a");
    }while(ch!='s');

   j=0;
   if(Kp==0 && Kd==0 && Ki==0)
     {
      gotoxy(1,17);
      puts("                                  ");
      puts("\a");
      puts("TODAS LAS CONSTANTES NO PUEDEN SER CEROS");
      j=1;
     }
  }while(j==1);

 inicio=factor_dato/(float)longit_vect;

 for(i=0;i<longit_vect;i++)
   {
    if(i<80)
     w=inicio*(i+1);
    if(i>=80 && i<=190)
     w=0.4*i-27.6;
    if(i>190)
     w=2*i-331.6;

    realFTc[i] =Kp*realFT[i]+(Kp*Ki/w-Kp*Kd*w)*imagFT[i];
    imagFTc[i] =(Kp*Kd*w-Kp*Ki/w)*realFT[i]+Kp*imagFT[i];
   }

 hayGc=1;

 graphicsStop();
 resume();
}



int TVAesfudes::RecuperaGc()
{
```

```
char archivo[30];
float dato;
FILE *fpin;
int i, j, k;
float  w, inicio;

for(i=0;i<30;i++)
  archivo[i]='\0';

i=0;
TDialog *abrir = new TDialog(TRect(0, 0, 40, 13), "Recuperar una Gc");

abrir->insert(
  new TStaticText(TRect(5, 2, 35, 8),
    "\003UTILIZAR UNA Gc GRABADA\n"
    "\003ANTERIORMENTE\n\003\n"
    "\003Ingrese el nombre\n"
    "\003del archivo de entrada\n"
    "\003(sin extensión)"
    )
  );

abrir->insert(new TInputLine(TRect(10,8,30,9),30));

abrir->insert(
  new TButton(TRect(7, 10, 17, 12), "~O~K", cmOK, bfDefault)
  );
abrir->insert(
  new TButton(TRect(23, 10, 33, 12), "~C~ancel", cmCancel, bfNormal)
  );
abrir->options |= ofCentered;

abrir->setData(archivo);

ushort control=deskTop->execView(abrir);
k=0;
if(control == cmOK)
  {
    i=1;
    k=1;
    abrir->getData(archivo);
  }
destroy(abrir);


if(i==1)
  {
    /*INICIALIZACION DE ARREGLOS*/
    for(i=0;i<200;i++)
      realFTc[i]=0;
    for(i=0;i<200;i++)
      imagFTc[i]=0;

    for(i=1;i<30;i++)
     {
       if(archivo[i]=='\0')
         {
           j=i;
           i=30;
         }
     }

    archivo[j]='.';
    archivo[j+1]='g';
    archivo[j+2]='c';
    archivo[j+3]='\0';

    if((fpin=fopen(archivo,"r"))==NULL)
     {
```

```
        puts("\a");
        messageBox("No se puede leer el archivo",
         mfError | mfOKButton);
        return 0;
      }

    fscanf(fpin,"%f",&Kp);
    fscanf(fpin,"%f",&Kd);
    fscanf(fpin,"%f",&Ki);

    }

if (k==1)
  {
    inicio=factor_dato/(float)longit_vect;

    for(i=0;i<longit_vect;i++)
     {
       if(i<80)
         w=inicio*(i+1);
       if(i>=80 && i<=190)
         w=0.4*i-27.6;
       if(i>190)
         w=2*i-331.6;

       realFTc[i] =Kp*realFT[i]+(Kp*Ki/w-Kp*Kd*w)*imagFT[i];
       imagFTc[i] =(Kp*Kd*w-Kp*Ki/w)*realFT[i]+Kp*imagFT[i];
     }

    hayGc=1;
  }

return 0;
}
```

```
/*-------------------------------------------------------*/
/*                                                       */
/*   PROGRAMA PARA ANALIZAR ESTABILIDAD                  */
/*   UTILIZANDO LA FUNCION DESCRIPTIVA                   */
/*                                                       */
/*     FUNCION DESCRIPTIVA (FIGURAS)                     */
/*                                                       */
/*   Wilmer Cárdenas Burbano 1994                        */
/*-------------------------------------------------------*/

#define Uses_MsgBox
#define Uses_TApplication

#include <tv.h>
#include <string.h>
#include <graphics.h>

#include "funcion.h"



                /*R E L E S*/

/*RELE IDEAL*/
int TVAesfudes::fig_rele_Ideal()
{
    int xmax, i;
    const MAXSIZE = 80;
    char errorMsg[MAXSIZE];
    suspend();

    i=1;
    if (graphicsStart() == False)
      {
       strcpy(errorMsg,grapherrormsg(graphresult()));
       strcat( errorMsg,"." );
       messageBox(errorMsg, mfError | mfOKButton);
       i=0;
      }
    else
      {
       setbkcolor(BLUE);
       xmax=getmaxx();
       cleardevice();
       setcolor(6);
       setlinestyle(0,0,0);
       line(xmax/2,10,xmax/2,190);          /*Eje Y*/
       line(xmax/2-90,100,xmax/2+90,100);   /*Eje X*/
       setcolor(15);
       setlinestyle(0,0,3);
       settextjustify(1,1);

       outtextxy(xmax/2,5,"CASO DE UN RELE IDEAL");
       line(xmax/2-80,160,xmax/2,160);  /*Linea inferior*/
       line(xmax/2,160,xmax/2,40);      /*Linea abajo-arriba*/
       line(xmax/2,40,xmax/2+80,40);    /*Linea superior*/
       outtextxy(xmax/2-12,40,"+U");    /*Nombre superior*/
       outtextxy(xmax/2+12,160,"-U");   /*Nombre inferior*/
       setcolor(YELLOW);
      }
    return i;
}



 /*RELE CON ZONA MUERTA*/
int TVAesfudes::fig_rele_ZM()
{
```

```
    int i, xmax;
    const MAXSIZE = 80;
    char errorMsg[MAXSIZE];
    suspend();

    i=1;
    if (graphicsStart() == False)
      {
       strcpy(errorMsg,grapherrormsg(graphresult()));
       strcat( errorMsg,"." );
       messageBox(errorMsg, mfError | mfOKButton);
       i=0;
      }
    else
      {
       setbkcolor(BLUE);
       settextjustify(1,1);
       xmax=getmaxx();
       cleardevice();
       setcolor(6);
       setlinestyle(0,0,0);
       line(xmax/2,10,xmax/2,190);          /*Eje Y*/
       line(xmax/2-90,100,xmax/2+90,100);   /*Eje X*/
       setcolor(15);
       setlinestyle(0,0,3);
       settextjustify(1,1);
       outtextxy(xmax/2,5,"CASO DE UN RELE CON ZONA MUERTA");
       line(xmax/2-80,160,xmax/2-25,160);   /*Linea inferior*/
       line(xmax/2-25,160,xmax/2-25,100);   /*Linea abajo-medio*/
       line(xmax/2-25,100,xmax/2+25,100);   /*Linea eje X*/
       line(xmax/2+25,100,xmax/2+25,40);    /*Linea medio-arriba*/
       line(xmax/2+25,40, xmax/2+80,40);    /*Linea superior*/
       outtextxy(xmax/2-12,40,"+U");        /*Nombre superior*/
       outtextxy(xmax/2+12,160,"-U");       /*Nombre inferior*/
       outtextxy(xmax/2+25,110,"+D");       /*Nombre Derecha*/
       outtextxy(xmax/2-25,90,"-D");        /*Nombre Izquierda*/
       setcolor(YELLOW);
      }
    return i;
}


/*RELE CON PENDIENTE*/
int TVAesfudes::fig_rele_Pend()
{
    int i, xmax;
    const MAXSIZE = 80;
    char errorMsg[MAXSIZE];
    suspend();

    i=1;
    if (graphicsStart() == False)
      {
       strcpy(errorMsg,grapherrormsg(graphresult()));
       strcat( errorMsg,"." );
       messageBox(errorMsg, mfError | mfOKButton);
       i=0;
      }
    else
      {
       setbkcolor(BLUE);
       settextjustify(1,1);
       xmax=getmaxx();
       cleardevice();
       setcolor(6);
       setlinestyle(0,0,0);
       line(xmax/2,10,xmax/2,190);          /*Eje Y*/
       line(xmax/2-90,100,xmax/2+90,100);   /*Eje X*/
```

```
        setcolor(15);
        setlinestyle(0,0,3);
        settextjustify(1,1);

        outtextxy(xmax/2,5,"CASO DE UN RELE CON PENDIENTE");
        line(xmax/2-80,165,xmax/2,160);       /*Pend. inferior*/
        line(xmax/2,160,    xmax/2,40);       /*Linea abajo-arriba*/
        line(xmax/2,40,     xmax/2+80,35);    /*Pend. Superior*/
        outtextxy(xmax/2-12,40,"+U");         /*Nombre superior*/
        outtextxy(xmax/2+12,160,"-U");        /*Nombre inferior*/
        outtextxy(xmax/2+25,33,"K");          /*Pendiente sup.*/
        outtextxy(xmax/2-25,153,"K");         /*Pend.inferior*/
        setcolor(YELLOW);
      }
    return i;
}



//RELE CON ZONA MUERTA Y PENDIENTE
int TVAesfudes::fig_rele_ZMyP()
{
    int i, xmax;
    const MAXSIZE = 80;
    char errorMsg[MAXSIZE];
    suspend();

    i=1;

    if (graphicsStart() == False)
      {
        strcpy(errorMsg,grapherrormsg(graphresult()));
        strcat( errorMsg,"." );
        messageBox(errorMsg, mfError | mfOKButton);
        i=0;
      }
    else
      {
        setbkcolor(BLUE);
        settextjustify(1,1);
        xmax=getmaxx();
        cleardevice();
        setcolor(6);
        setlinestyle(0,0,0);
        line(xmax/2,10,xmax/2,190);           /*Eje Y*/
        line(xmax/2-90,100,xmax/2+90,100);    /*Eje X*/
        setcolor(15);
        setlinestyle(0,0,3);
        settextjustify(1,1);

        outtextxy(xmax/2,5,"CASO DE UN RELE CON ZONA MUERTA Y PENDIENTE");
        line(xmax/2-80,165,xmax/2-25,160);    /*Pend. inferior*/
        line(xmax/2-25,160,xmax/2-25,100);    /*Linea abajo-medio*/
        line(xmax/2-25,100,xmax/2+25,100);    /*Linea eje X*/
        line(xmax/2+25,100,xmax/2+25,40);     /*Linea medio-arriba*/
        line(xmax/2+25,40, xmax/2+80,35);     /*Pend. superior*/
        outtextxy(xmax/2-12,40,"+U");         /*Nombre superior*/
        outtextxy(xmax/2+12,160,"-U");        /*Nombre inferior*/
        outtextxy(xmax/2+25,110,"+D");        /*Nombre Derecha*/
        outtextxy(xmax/2-25,90,"-D");         /*Nombre Izquierda*/
        outtextxy(xmax/2+50,33,"K");          /*Pendiente sup.*/
        outtextxy(xmax/2-50,153,"K");         /*Pend.inferior*/
        setcolor(YELLOW);
      }
    return i;
}
```

```
                /*S A T U R A C I O N*/

/*SATURACION IDEAL*/
int TVAesfudes::fig_sat_ideal()
{
    int i, xmax;
    const MAXSIZE = 80;
    char errorMsg[MAXSIZE];
    suspend();

    i=1;

    if (graphicsStart() == False)
      {
       strcpy(errorMsg,grapherrormsg(graphresult()));
       strcat( errorMsg,"." );
       messageBox(errorMsg, mfError | mfOKButton);
       i=0;
      }
    else
      {
       setbkcolor(BLUE);
       xmax=getmaxx();
       cleardevice();
       setcolor(6);
       setlinestyle(0,0,0);
       line(xmax/2,10,xmax/2,190);          /*Eje Y*/
       line(xmax/2-90,100,xmax/2+90,100);   /*Eje X*/
       setcolor(15);
       setlinestyle(0,0,3);
       settextjustify(1,1);

       outtextxy(xmax/2,5,"CASO DE UNA SATURACION IDEAL");
       line(xmax/2-80,160,xmax/2-25,160);   /*Linea inferior*/
       line(xmax/2-25,160,xmax/2+25,40);    /*Pend. abajo-arriba*/
       line(xmax/2+25,40, xmax/2+80,40);    /*Linea superior*/
       outtextxy(xmax/2-12,40,"+U");        /*Nombre superior*/
       outtextxy(xmax/2+12,160,"-U");       /*Nombre inferior*/
       outtextxy(xmax/2+25,110,"+D");
       outtextxy(xmax/2-25,90,"-D");
       outtextxy(xmax/2-2,70,"K");          /*Nombre de la pendiente*/
       setcolor(YELLOW);
      }
    return i;
}



 /*SATURACION CON ZONA MUERTA*/
int TVAesfudes::fig_sat_ZM()
{

    int i, xmax;
    const MAXSIZE = 80;
    char errorMsg[MAXSIZE];
    suspend();

    i=1;

    if (graphicsStart() == False)
      {
       strcpy(errorMsg,grapherrormsg(graphresult()));
       strcat( errorMsg,"." );
       messageBox(errorMsg, mfError | mfOKButton);
       i=0;
```

```
    }
  else
    {
     setbkcolor(BLUE);
     xmax=getmaxx();
     cleardevice();
     setcolor(6);
     setlinestyle(0,0,0);
     line(xmax/2,10,xmax/2,190);          /*Eje Y*/
     line(xmax/2-90,100,xmax/2+90,100);   /*Eje X*/
     setcolor(15);
     setlinestyle(0,0,3);
     settextjustify(1,1);

     outtextxy(xmax/2,5,"CASO DE UNA SATURACION CON ZONA MUERTA");
     line(xmax/2-80,160,xmax/2-30,160);   /*Linea inferior*/
     line(xmax/2-30,160,xmax/2-15,100);   /*Pend. abajo-medio*/
     line(xmax/2-15,100,xmax/2+15,100);   /*Linea eje X*/
     line(xmax/2+15,100,xmax/2+30,40);    /*Pend. medio-arriba*/
     line(xmax/2+30,40, xmax/2+80,40);    /*Linea superior*/
     outtextxy(xmax/2-12,40,"+U");        /*Nombre superior*/
     outtextxy(xmax/2+12,160,"-U");       /*Nombre inferior*/
     outtextxy(xmax/2+30,120,"+D2");
     outtextxy(xmax/2-30,80,"-D2");
     outtextxy(xmax/2+15,110,"+D1");
     outtextxy(xmax/2-15,90,"-D1");
     outtextxy(xmax/2+10,70,"K");          /*Nombre de la pendiente*/
     outtextxy(xmax/2-10,130,"K");
     setcolor(YELLOW);
    }
  return i;
}



/*SATURACION CON PENDIENTE*/
int TVAesfudes::fig_sat_Pend()
 {
   int i, xmax;
   const MAXSIZE = 80;
   char errorMsg[MAXSIZE];
   suspend();

   i=1;

   if (graphicsStart() == False)
     {
      strcpy(errorMsg,grapherrormsg(graphresult()));
      strcat( errorMsg,"." );
      messageBox(errorMsg, mfError | mfOKButton);
      i=0;
     }
   else
     {
      setbkcolor(BLUE);
      xmax=getmaxx();
      cleardevice();
      setcolor(6);
      setlinestyle(0,0,0);
      line(xmax/2,10,xmax/2,190);          /*Eje Y*/
      line(xmax/2-90,100,xmax/2+90,100);   /*Eje X*/
      setcolor(15);
      setlinestyle(0,0,3);
      settextjustify(1,1);

      outtextxy(xmax/2,5,"CASO DE UNA SATURACION CON PENDIENTE");
      line(xmax/2-80,165,xmax/2-25,160);   /*Linea inferior*/
      line(xmax/2-25,160,xmax/2+25,40);    /*Linea abajo-arriba*/
```

```
        line(xmax/2+25,40, xmax/2+80,35);     /*Linea superior*/
        outtextxy(xmax/2-12,40,"+U");         /*Nombre superior*/
        outtextxy(xmax/2+12,160,"-U");        /*Nombre inferior*/
        outtextxy(xmax/2+25,110,"+D");
        outtextxy(xmax/2-25,110,"-D");
        outtextxy(xmax/2+40,30,"K");          /*Nombre de la pendiente*/
        setcolor(YELLOW);
      }
    return 1;
}



/*SATURACION CON ZONA MUERTA Y PENDIENTE*/
int TVAesfudes::fig_sat_ZMyP()
{
    int i, xmax;
    const MAXSIZE = 80;
    char errorMsg[MAXSIZE];
    suspend();

    i=1;

    if (graphicsStart() == False)
      {
       strcpy(errorMsg,grapherrormsg(graphresult()));
       strcat( errorMsg,"." );
       messageBox(errorMsg, mfError | mfOKButton);
       i=0;
      }
    else
      {
       setbkcolor(BLUE);
       xmax=getmaxx();
       cleardevice();
       setcolor(6);
       setlinestyle(0,0,0);
       line(xmax/2,10,xmax/2,190);            /*Eje Y*/
       line(xmax/2-90,100,xmax/2+90,100);     /*Eje X*/
       setcolor(15);
       setlinestyle(0,0,3);
       settextjustify(1,1);

       outtextxy(xmax/2,5,"CASO DE UNA SATURACION CON ZONA MUERTA Y PENDIENTE");
       line(xmax/2-80,165,xmax/2-30,160);     /*Linea inferior*/
       line(xmax/2-30,160,xmax/2-15,100);     /*Pend. abajo-medio*/
       line(xmax/2-15,100,xmax/2+15,100);     /*Linea eje X*/
       line(xmax/2+15,100,xmax/2+30,40);      /*Pend. medio-arriba*/
       line(xmax/2+30,40, xmax/2+80,35);      /*Linea superior*/
       outtextxy(xmax/2-12,40,"+U");          /*Nombre superior*/
       outtextxy(xmax/2+12,160,"-U");         /*Nombre inferior*/
       outtextxy(xmax/2+30,120,"+D2");
       outtextxy(xmax/2-30,80,"-D2");
       outtextxy(xmax/2+15,110,"+D1");
       outtextxy(xmax/2-15,90,"-D1");
       outtextxy(xmax/2+40,30,"K");           /*Nombre de la pendiente*/
       outtextxy(xmax/2-40,170,"K");
       setcolor(YELLOW);
      }
    return 1;
}


                /*H I S T E R E S I S*/

/*HISTERESIS IDEAL*/
```

```
int TVAesfudes::fig_hist_Ideal()
 {
    int i, xmax;
    const MAXSIZE = 80;
    char errorMsg[MAXSIZE];
    suspend();

    i=1;

    if (graphicsStart() == False)
      {
       strcpy(errorMsg,grapherrormsg(graphresult()));
       strcat( errorMsg,"." );
       messageBox(errorMsg, mfError | mfOKButton);
       i=0;
      }
    else
      {
       setbkcolor(BLUE);
       xmax=getmaxx();
       cleardevice();
       setcolor(6);
       setlinestyle(0,0,0);
       line(xmax/2,10,xmax/2,190);          /*Eje Y*/
       line(xmax/2-90,100,xmax/2+90,100);   /*Eje X*/
       setcolor(15);
       setlinestyle(0,0,3);
       settextjustify(1,1);

       outtextxy(xmax/2,5,"CASO DE UNA HISTERESIS IDEAL");

       line(xmax/2-80,160,xmax/2+20,160);   /*Linea inferior*/
       line(xmax/2+20,160,xmax/2+20,40);    /*Linea abajo-arriba2*/
       line(xmax/2-20,160,xmax/2-20,40);    /*Linea abajo-arriba1*/
       line(xmax/2-20,40,xmax/2+80,40);     /*Linea superior*/

       outtextxy(xmax/2-35,40,"+U");        /*Nombre superior*/
       outtextxy(xmax/2+35,160,"-U");       /*Nombre inferior*/
       outtextxy(xmax/2+30,110,"+D");
       outtextxy(xmax/2-30,90,"-D");
       setcolor(YELLOW);
      }
    return i;
 }



 /*HISTERESIS CON ZONA MUERTA*/
int TVAesfudes::fig_hist_ZM()
 {
    int i, xmax;
    const MAXSIZE = 80;
    char errorMsg[MAXSIZE];
    suspend();

    i=1;

    if (graphicsStart() != False)
      {
       strcpy(errorMsg,grapherrormsg(graphresult()));
       strcat( errorMsg,"." );
       messageBox(errorMsg, mfError | mfOKButton);
       i=0;
      }
    else
      {
       setbkcolor(BLUE);
       xmax=getmaxx();
```

```
      cleardevice();
      setcolor(6);
      setlinestyle(0,0,0);
      line(xmax/2,10,xmax/2,190);          /*Eje Y*/
      line(xmax/2-90,100,xmax/2+90,100);   /*Eje X*/
      setcolor(15);
      setlinestyle(0,0,3);
      settextjustify(1,1);

      outtextxy(xmax/2,5,"CASO DE UNA HISTERESIS CON ZONA MUERTA");

      line(xmax/2-80,160,xmax/2-20,160);   /*Linea inferior*/
      line(xmax/2-20,160,xmax/2-20,100);   /*Linea abajo-medio2*/
      line(xmax/2-50,160,xmax/2-50,100);   /*Linea abajo-medio1*/
      line(xmax/2-50,100,xmax/2+50,100);   /*Linea eje X*/
      line(xmax/2+50,100,xmax/2+50,40);    /*Linea medio_arriba2*/
      line(xmax/2+20,100,xmax/2+20,40);    /*Linea medio_arriba1*/
      line(xmax/2+20,40, xmax/2+80,40);    /*Linea superior*/

      outtextxy(xmax/2-12,40,"+U");        /*Nombre superior*/
      outtextxy(xmax/2+12,160,"-U");       /*Nombre inferior*/
      outtextxy(xmax/2+20,110,"+D1");
      outtextxy(xmax/2-20,90,"-D1");
      outtextxy(xmax/2+50,120,"+D2");
      outtextxy(xmax/2-50,80,"-D2");
      setcolor(YELLOW);
    }
  return i;
}



/*HISTERESIS CON PENDIENTE*/
int TVAesfudes::fig_hist_Pend()
{
  int i, xmax;
  const MAXSIZE = 80;
  char errorMsg[MAXSIZE];
  suspend();

  i=1;

  if (graphicsStart() == False)
    {
      strcpy(errorMsg,grapherrormsg(graphresult()));
      strcat( errorMsg,"." );
      messageBox(errorMsg, mfError | mfOKButton);
      i=0;
    }
  else
    {
      setbkcolor(BLUE);
      xmax=getmaxx();
      cleardevice();
      setcolor(6);
      setlinestyle(0,0,0);
      line(xmax/2,10,xmax/2,190);          /*Eje Y*/
      line(xmax/2-90,100,xmax/2+90,100);   /*Eje X*/
      setcolor(15);
      setlinestyle(0,0,3);
      settextjustify(1,1);

      outtextxy(xmax/2,5,"CASO DE UNA HISTERESIS CON PENDIENTE");
      line(xmax/2-80,160,xmax/2-20,160);   /*Linea inferior*/
      line(xmax/2-20,160,xmax/2+50,40);    /*Linea abajo-arriba2*/
      line(xmax/2-50,160,xmax/2+20,40);    /*Linea abajo-arriba1*/
      line(xmax/2+20,40,xmax/2+80,40);     /*Linea superior*/
      outtextxy(xmax/2-12,40,"+U");        /*Nombre superior*/
```

```
      outtextxy(xmax/2+12,160,"-U");        /*Nombre inferior*/
      outtextxy(xmax/2+20,110," +D1");
      outtextxy(xmax/2-20,90,"-D1");
      outtextxy(xmax/2+50,120,"+D2");
      outtextxy(xmax/2-50,80,"-D2");
      outtextxy(xmax/2-5,70,"K");
      setcolor(YELLOW);
    }
  return 1;
}



/*HISTERESIS CON ZONA MUERTA Y PENDIENTE*/
int TVAesfudes::fig_hist_ZMyP()
{
  int i, xmax;
  const MAXSIZE = 80;
  char errorMsg[MAXSIZE];
  suspend();

  i=1;

  if (graphicsStart() == False)
    {
      strcpy(errorMsg,grapherrormsg(graphresult()));
      strcat( errorMsg,"." );
      messageBox(errorMsg, mfError | mfOKButton);
      i=0;
    }
  else
    {
      setbkcolor(BLUE);
      xmax=getmaxx();
      cleardevice();
      setcolor(6);
      setlinestyle(0,0,0);
      line(xmax/2,10,xmax/2,190);           /*Eje Y*/
      line(xmax/2-90,100,xmax/2+90,100);    /*Eje X*/
      setcolor(15);
      setlinestyle(0,0,3);
      settextjustify(1,1);

      outtextxy(xmax/2,5,"CASO DE UNA HISTERESIS CON ZONA MUERTA Y PENDIENTE");
      line(xmax/2-80,160,xmax/2-40,160);    /*Linea inferior*/
      line(xmax/2-40,160,xmax/2-20,100);    /*Linea abajo-medio2*/
      line(xmax/2-60,160,xmax/2-40,100);    /*Linea abajo-medio1*/
      line(xmax/2-40,100,xmax/2+40,100);    /*Linea eje X*/
      line(xmax/2+40,100,xmax/2+60,40);     /*Linea medio_arriba2*/
      line(xmax/2+20,100,xmax/2+40,40);     /*Linea medio_arriba1*/
      line(xmax/2+40,40, xmax/2+80,40);     /*Linea superior*/
      outtextxy(xmax/2-12,40,"+U");         /*Nombre superior*/
      outtextxy(xmax/2+12,160,"-U");        /*Nombre inferior*/
      outtextxy(xmax/2+20,110," +D1");
      outtextxy(xmax/2-20,90,"-D1");
      outtextxy(xmax/2+40,120,"+D2");
      outtextxy(xmax/2-40,80,"-D2");
      outtextxy(xmax/2+60,130,"+D3");
      outtextxy(xmax/2-60,70,"-D3");
      outtextxy(xmax/2+80,140,"+D4");
      outtextxy(xmax/2-80,60,"-D4");
      outtextxy(xmax/2+20,70,"K");
      outtextxy(xmax/2-20,130,"K");
      setcolor(YELLOW);
    }
  return 1;
}
```

```
/*-------------------------------------------------------*/
/*                                                       */
/*   PROGRAMA PARA ANALIZAR ESTABILIDAD                  */
/*   UTILIZANDO LA FUNCION DESCRIPTIVA                   */
/*                                                       */
/*     FUNCION DESCRIPTIVA (RELES)                       */
/*                                                       */
/*   Wilmer Cárdenas Burbano 1994                        */
/*-------------------------------------------------------*/

#define Uses_MsgBox
#define Uses_TApplication

#define pi 3.141592

#include <tv.h>
#include <stdio.h>
#include <math.h>
#include <graphics.h>
#include <conio.h>
#include <ctype.h>

#include "funcion.h"

extern float realFD[];
extern float imagFD[];
extern int num_datosFD;
extern float inicioAFD;
extern int tipoNL;
extern float pto1, pto2, pto3;


/*FUNCIONES PARA CALCULAR LAS COORDENADAS DE LA FUNCION DESCRIPTIVA*/
/*CASO DE UN RELE IDEAL*/
int TVAesfudes::rele_Ideal()
{
    int i;
    float A, U;
    char ch;

    /*INICIALIZACION DE ARREGLOS*/
    for(i=0;i<60;i++)
      realFD[i]=0;
    for(i=0;i<60;i++)
      imagFD[i]=0;

    do
      {
       do
         {
          i=1;
          gotoxy(1,15);
          puts("Ingrese el valor de U");
          gotoxy(30,15);
          puts("U= ");
          gotoxy(33,15);
          fflush(stdin);
          while(scanf("%f",&U)==0)
            {
             fflush(stdin);
             gotoxy(33,15);
             printf("\a          ");
             gotoxy(1,16);
             puts("                                    ");
             gotoxy(33,15);
            }
          gotoxy(1,16);
          puts("                                    ");
```

```
        printf("%f",U);

        if(U<-0)
          {
           gotoxy(1,16);
           setcolor(RED);
           puts("\aU DEBE SER MAYOR QUE CERO");
           setcolor(YELLOW);
           gotoxy(30,15);
           puts("            ");
           i=0;
          }
      }while(i--0);

    gotoxy(1,16);
    puts("                                ");
    gotoxy(1,17);
    setcolor(RED);
    printf("LOS DATOS SON CORRECTOS s/n\t");
    setcolor(YELLOW);
    fflush(stdin);
    ch=getch();
    if(isupper(ch))
      ch=tolower(ch);
    if(ch!='s')
      {
        gotoxy(1,15);
        puts("\a                                              ");
        gotoxy(1,17);
        puts("                                            ");
      }
  }while(ch!='s');

  /*CALCULO DE -1/N*/
  inicioAFD=0.001;
  for(i=0;i<num_datosFD;i++)
    {
     if(i<=20)
       A=0.3*i+inicioAFD;
     else
       A=2*i-34+inicioAFD;

     realFD[i]=-pi*A/(4*U);
     imagFD[i]=0;
    }

  tipoNL=1;
  ptoi=U;
  settextjustify(0,0);
  graphicsStop();
  resume();
  return inicioAFD;
}



 /*CASO DE UN RELE CON ZONA MUERTA*/
int TVAesfudes::rele_ZM()
{
  int i;
  float U, D, A;
  double tet;
  char ch;

  /*INICIALIZACION DE ARREGLOS*/
  for(i=0;i<60;i++)
    realFD[i]=0;
```

```
for(i=0;i<50;i++)
  imagFD[i]=0;

do
  {
    do
      {
        i=1;
        gotoxy(1,15);
        puts("Ingrese el valor de U");
        gotoxy(30,15);
        puts("U= ");
        gotoxy(33,15);
        fflush(stdin);
        while(scanf("%f",&U)==0)
          {
            fflush(stdin);
            gotoxy(33,15);
            printf("\a          ");
            gotoxy(1,16);
            puts("                                        ");
            gotoxy(33,15);
          }
        gotoxy(1,16);
        puts("                                      ");
        gotoxy(33,15);
        printf("%f",U);

        if(U<=0)
          {
            gotoxy(1,16);
            setcolor(RED);
            puts("\aU DEBE SER MAYOR QUE CERO");
            setcolor(YELLOW);
            gotoxy(30,15);
            puts("            ");
            i=0;
          }
      }while(i==0);

    do
      {
        i=1;
        gotoxy(1,16);
        puts("                                    ");
        gotoxy(1,17);
        puts("Ingrese el valor de D");
        gotoxy(30,17);
        puts("D= ");
        fflush(stdin);
        gotoxy(33,17);
        while(scanf("%f",&D)==0)
          {
            fflush(stdin);
            gotoxy(33,17);
            puts("\a             ");
            gotoxy(1,18);
            puts("                                        ");
            gotoxy(33,17);
          }
        gotoxy(1,18);
        puts("                                      ");
        gotoxy(33,17);
        printf("%f",D);

        if(D<=0)
          {
            gotoxy(1,18);
```

```
            puts("\aD DEBE SER MAYOR QUE CERO");
            setcolor(YELLOW);
            gotoxy(30,17);
            puts("            ");
            i=0;
          }
      }while(i--0);

      gotoxy(1,18);
      puts("                                ");
      gotoxy(1,19);
      setcolor(RED);
      printf("LOS DATOS SON CORRECTOS s/n\t");
      setcolor(YELLOW);
      fflush(stdin);
      ch=getch();
      if(isupper(ch))
      ch=tolower(ch);
      if(ch!='s')
        {
          gotoxy(1,15);
          puts("\a                                          ");
          gotoxy(1,17);
          puts("                                        ");
          gotoxy(1,19);
          puts("                                        ");
        }
    }while(ch!='s');

  /*CALCULO DE -1/N*/
  inicioAFD=D+0.001;
  for(i=0;i<num_datosFD;i++)
    {
     if(i<=20)
       A=0.3*i+inicioAFD;
     else
       A=2*i-34+inicioAFD;

     tet=asin((double)D/A);
     realFD[i]=-pi*(A)/(4*U*cos(tet));
     imagFD[i]=0;
    }

  tipoNL=2;
  pto1=U;
  pto2=D;
  settextjustify(0,0);
  graphicsStop();
  resume();
  return inicioAFD;
}



/*CASO DE UN RELE CON PENDIENTE*/
int IVAesfudes::rele_Pend()
{
  int i;
  float A, U, K;
  char ch;

  /*INICIALIZACION DE ARREGLOS*/
  for(i=0;i<60;i++)
    realFD[i]=0;
  for(i=0;i<60;i++)
    imagFD[i]=0;

  do
```

```
{
 do
   {
     i=1;
     gotoxy(1,15);
     puts("Ingrese el valor de U");
     gotoxy(30,15);
     puts("U= ");
     gotoxy(33,15);
     fflush(stdin);
     while(scanf("%f",&U)==0)
       {
        fflush(stdin);
        gotoxy(33,15);
        printf("\a         ");
        gotoxy(1,16);
        puts("                                        ");
        gotoxy(33,15);
       }
     gotoxy(1,16);
     puts("                                        ");
     gotoxy(33,15);
     printf("%f",U);

     if(U<=0)
       {
        gotoxy(1,16);
        setcolor(RED);
        puts("\aU DEBE SER MAYOR QUE CERO");
        setcolor(YELLOW);
        gotoxy(30,15);
        puts("            ");
        i=0;
       }
   }while(i--0);

 gotoxy(1,16);
 puts("                                    ");
 gotoxy(1,17);
 puts("Ingrese el valor de K");
 gotoxy(30,17);
 puts("K= ");
 fflush(stdin);
 gotoxy(33,17);
 while(scanf("%f",&K)==0)
   {
     fflush(stdin);
     gotoxy(33,17);
     puts("\a             ");
     gotoxy(1,18);
     puts("                                        ");
     gotoxy(33,17);
   }
 gotoxy(1,18);
 puts("                                      ");
 gotoxy(33,17);
 printf("%f",K);

 gotoxy(1,19);
 setcolor(RED);
 printf("LOS DATOS SON CORRECTOS s/n\t");
 setcolor(YELLOW);
 fflush(stdin);
 ch=getch();
 if(isupper(ch))
   ch=tolower(ch);
 if(ch!='s')
   {
     gotoxy(1,15);
```

```
            puts("\a                                                    ");
            gotoxy(1,17);
            puts("                                                  ");
            gotoxy(1,19);
            puts("                                                  ");
          }
      }while(ch!='s');

    /*CALCULO DE -1/N*/
    inicioAFD=0.001;
    for(i=0;i<num_datosFD;i++)
      {
       if(i<=20)
         A=0.3*i+inicioAFD;
       else
         A=2*i-34+inicioAFD;

       realFD[i]=-pi*(A)/(4*U+K*pi*A);
       imagFD[i]=0;
      }

    tipoNL=3;
    pto1=U;
    pto2=K;
    settextjustify(0,0);
    graphicsStop();
    resume();
    return inicioAFD;
}



int TVAesfudes::rele_ZMyP()
{
    int i;
    float A, U, D, K, N;
    double tet;
    char ch;

    /*INICIALIZACION DE ARREGLOS*/
    for(i=0;i<60;i++)
      realFD[i]=0;
    for(i=0;i<60;i++)
      imagFD[i]=0;

    do
      {
       do
         {
           i=1;
           gotoxy(1,15);
           puts("Ingrese el valor de U");
           gotoxy(30,15);
           puts("U= ");
           gotoxy(33,15);
           fflush(stdin);
           while(scanf("%f",&U)==0)
             {
              fflush(stdin);
              gotoxy(33,15);
              printf("\a         ");
              gotoxy(1,16);
              puts("                                          ");
              gotoxy(33,15);
             }
           gotoxy(1,16);
```

```
    puts("                                        ");
    gotoxy(33,15);
    printf("%f",U);

    if(U<-0)
      {
       gotoxy(1,16);
       setcolor(RED);
       puts("\aU DEBE SER MAYOR QUE CERO");
       setcolor(YELLOW);
       gotoxy(33,15);
       puts("            ");
       i-0;
      }
  }while(i--0);

do
  {
    gotoxy(1,16);
    puts("                                      ");
    i-1;
    gotoxy(1,17);
    puts("Ingrese el valor de D");
    gotoxy(30,17);
    puts("D= ");
    fflush(stdin);
    gotoxy(33,17);
    while(scanf("%f",&D)--0)
      {
       fflush(stdin);
       gotoxy(33,17);
       puts("\a             ");
       gotoxy(1,18);
       puts("                                        ");
       gotoxy(33,17);
      }
    gotoxy(1,18);
    puts("                                      ");
    gotoxy(33,17);
    printf("%f",D);

    if(D<-0)
      {
       gotoxy(1,18);
       setcolor(RED);
       puts("\aD DEBE SER MAYOR QUE CERO");
       setcolor(YELLOW);
       gotoxy(30,17);
       puts("            ");
       i-0;
      }
  }while(i--0);

gotoxy(1,18);
puts("                                          ");
gotoxy(1,19);
puts("Ingrese el valor de K");
gotoxy(30,19);
puts("K= ");
gotoxy(33,19);
fflush(stdin);
while(scanf("%f",&K)==0)
  {
    fflush(stdin);
    gotoxy(33,19);
    printf("\a         ");
    gotoxy(1,20);
    puts("                                      ");
    gotoxy(33,19);
```

```
        }
      gotoxy(1,20);
      puts("                                        ");
      gotoxy(33,19);
      printf("%f",K);

      gotoxy(1,21);
      setcolor(RED);
      printf("LOS DATOS SON CORRECTOS s/n\t");
      setcolor(YELLOW);
      fflush(stdin);
      ch=getch();
      if(isupper(ch))
      ch=tolower(ch);
      if(ch!='s')
        {
          gotoxy(1,15);
          puts("\a                                                ");
          gotoxy(1,17);
          puts("                                                ");
          gotoxy(1,19);
          puts("                                                ");
          gotoxy(1,21);
          puts("                                                ");
        }
    }while(ch!='s');

  /*CALCULO DE -1/N*/
  inicioAFD=D+0.001;
  for(i=0;i<num_datosFD;i++)
    {
      if(i<=20)
        A=0.3*i+inicioAFD;
      else
        A=2*i-34+inicioAFD;

      if (A==D)
        A=A+0.001;
      tet=asin((double)D/(double)A);
      N=(pi-2*tet+sin(2*tet))*K/pi + (4/(pi*A))*(U-K*D)*cos(tet);

      if(N==0)
        {
          graphicsStop();
          resume();
          puts("\a");
          messageBox("LOS DATOS SON INCONSISTENTES",
           mfError | mfOKButton);
          return 1;
        }
      realFD[i]=-1/N;
      imagFD[i]=0;
    }

  tipoNL=4;
  pto1=U;
  pto2=D;
  pto3=K;
  settextjustify(0,0);
  graphicsStop();
  resume();
  return inicioAFD;
}
```

```
/*-------------------------------------------------------*/
/*                                                       */
/*   PROGRAMA PARA ANALIZAR ESTABILIDAD                  */
/*   UTILIZANDO LA FUNCION DESCRIPTIVA                   */
/*                                                       */
/*   FUNCION DESCRIPTIVA (SATURACIONES)                  */
/*                                                       */
/*   Wilmer Cárdenas Burbano 1994                        */
/*-------------------------------------------------------*/


#define Uses_MsgBox
#define Uses_TApplication

#define pi 3.141592

#include <tv.h>
#include <stdio.h>
#include <math.h>
#include <graphics.h>
#include <conio.h>
#include <ctype.h>

#include "funcion.h"

extern float realFD[];
extern float imagFD[];
extern int num_datosFD;
extern float inicioAFD;
extern int tipoNL;
extern float pto1, pto2, pto3, pto4;


/*CASO DE UNA SATURACION IDEAL*/
int TVAesfudes::sat_Ideal()
{
    int i;
    float A, U, D, K, N;
    double tet;
    char ch;

    /*INICIALIZACION DE ARREGLOS*/
    for(i=0;i<60;i++)
      realFD[i]=0;
    for(i=0;i<60;i++)
      imagFD[i]=0;

    do
      {
       do
         {
          i=1;
          gotoxy(1,15);
          puts("Ingrese el valor de U");
          gotoxy(30,15);
          puts("U= ");
          gotoxy(33,15);
          fflush(stdin);
          while(scanf("%f",&U)==0)
            {
             fflush(stdin);
             gotoxy(33,15);
             printf("\a          ");
             gotoxy(1,16);
             puts("                                        ");
             gotoxy(33,15);
            }
          gotoxy(1,16);
          puts("                                        ");
```

```
      gotoxy(33,15);
      printf("%f",U);

      if(U<-0)
        {
         gotoxy(1,16);
         setcolor(RED);
         puts("\aU DEBE SER MAYOR QUE CERO");
         setcolor(YELLOW);
         gotoxy(33,15);
         puts("            ");
         i-0;
        }
    }while(i==0);

  gotoxy(1,16);
  puts("                                  ");
  do
    {
      i-1;
      gotoxy(1,17);
      puts("Ingrese el valor de D");
      gotoxy(30,17);
      puts("D- ");
      fflush(stdin);
      gotoxy(33,17);
      while(scanf("%f",&D)==0)
        {
         fflush(stdin);
         gotoxy(33,17);
         puts("\a             ");
         gotoxy(1,18);
         puts("                                      ");
         gotoxy(33,17);
        }
      gotoxy(1,18);
      puts("                                    ");
      gotoxy(33,17);
      printf("%f",D);

      if(D<-0)
        {
         gotoxy(1,18);
         setcolor(RED);
         puts("\aD DEBE SER MAYOR QUE CERO");
         setcolor(YELLOW);
         gotoxy(33,17);
         puts("            ");
         i-0;
        }
    }while(i==0);

  gotoxy(1,18);
  puts("                                    ");
  gotoxy(1,19);
  setcolor(RED);
  printf("LOS DATOS SON CORRECTOS s/n\t");
  setcolor(YELLOW);
      fflush(stdin);
  ch-getch();
  if(isupper(ch))
    ch=tolower(ch);
  if(ch!='s')
    {
      gotoxy(1,15);
      puts("\a                                          ");
      gotoxy(1,17);
      puts("                                        ");
      gotoxy(1,19);
```

```
          puts("                                                    ");
        }
    }while(ch!='s');

  /*CALCULO DE -1/N*/
  if(D==0)
    K=U/(D+0.001);
  else
    K=U/D;
  inicioAFD=D+0.001;
  for(i=0;i<num_datosFD;i++)
    {
     if(i<=20)
       A=0.3*i+inicioAFD;
     else
       A=2*i-34+inicioAFD;

     if(A==0)
     A=A+0.001;
     tet=asin((double)D/A);
     N=(2*tet-sin(2*tet))*K/pi+4U*cos(tet)/(pi*A);

     if(N==0)
       {
         graphicsStop();
         resume();
         puts("\a");
         messageBox("LOS DATOS SON INCONSISTENTES",
          mfError | mfOKButton);
         return i;
       }

     realFD[i]=-1/N;
     imagFD[i]=0;
    }

  tipoNL=5;
  pto1=U;
  pto2=D;
  settextjustify(0,0);
  graphicsStop();
  resume();
  return inicioAFD;
}



 /*CASO DE UNA SATURACION CON ZONA MUERTA*/
int TVAesfudes::sat_ZM()
{

  int i;
  float A, U, D1, D2, K, N;
  double tet1, tet2;
  char ch;

  /*INICIALIZACION DE ARREGLOS*/
  for(i=0;i<60;i++)
    realFD[i]=0;
  for(i=0;i<60;i++)
    imagFD[i]=0;

  do
    {
     do
       {
```

```
    gotoxy(1,15);
    puts("Ingrese el valor de U");
    gotoxy(30,15);
    puts("U= ");
    gotoxy(33,15);
    fflush(stdin);
    while(scanf("%f",&U)==0)
      {
       fflush(stdin);
       gotoxy(33,15);
       printf("\a          ");
       gotoxy(1,16);
       puts("                                        ");
       gotoxy(33,15);
      }
    gotoxy(1,16);
    puts("                                      ");
    gotoxy(33,15);
    printf("%f",U);

    if(U<=0)
      {
       gotoxy(1,16);
       setcolor(RED);
       puts("\aU DEBE SER MAYOR QUE CERO");
       setcolor(YELLOW);
       gotoxy(33,15);
       puts("            ");
       i=0;
      }
  }while(i==0);

gotoxy(1,16);
puts("                                ");
do
  {
    i=1;
    gotoxy(1,17);
    puts("Ingrese el valor de D1");
    gotoxy(30,17);
    puts("D1= ");
    fflush(stdin);
    gotoxy(33,17);
    while(scanf("%f",&D1)==0)
      {
       fflush(stdin);
       gotoxy(33,17);
       puts("\a              ");
       gotoxy(1,18);
       puts("                                        ");
       gotoxy(33,17);
      }
    gotoxy(1,18);
    puts("                                      ");
    gotoxy(33,17);
    printf("%f",D1);

    if(D1<=0)
      {
       gotoxy(1,18);
       setcolor(RED);
       puts("\aD1 DEBE SER MAYOR QUE CERO");
       setcolor(YELLOW);
       gotoxy(33,17);
       puts("            ");
       i=0;
      }
  }while(i==0);
```

```
    gotoxy(1,18);
    puts("                                          ");
    do
      {
        i=1;
        gotoxy(1,19);
        puts("Ingrese el valor de D2");
        gotoxy(30,19);
        puts("D2= ");
        gotoxy(33,19);
        fflush(stdin);
        while(scanf("%f",&D2)==0)
          {
            fflush(stdin);
            gotoxy(33,19);
            printf("\a          ");
            gotoxy(1,20);
            puts("                                      ");
            gotoxy(33,19);
          }
        gotoxy(1,20);
        puts("                                          ");
        gotoxy(33,19);
        printf("%f",D2);

        if(D2<=D1)
          {
            gotoxy(1,20);
            setcolor(RED);
            puts("\aD2 DEBE SER MAYOR QUE D1");
            setcolor(YELLOW);
            gotoxy(33,19);
            puts("             ");
            i=0;
          }
      }while(i--=0);

    gotoxy(1,20);
    puts("                                          ");
    gotoxy(1,21);
    setcolor(RED);
    printf("LOS DATOS SON CORRECTOS s/n\t");
    setcolor(YELLOW);
    fflush(stdin);
    ch=getch();
    if(isupper(ch))
    ch=tolower(ch);
    if(ch!='s')
      {
        gotoxy(1,15);
        puts("\a                                            ");
        gotoxy(1,17);
        puts("                                          ");
        gotoxy(1,19);
        puts("                                          ");
        gotoxy(1,21);
        puts("                                          ");
      }
  }while(ch!='s');


/*CALCULO DE -1/N*/
K=D2-D1;
if(K==0)
  K=K+0.001;
K=1/K;
inicioAFD=D2+0.001;
for(i=0;i<num_datosFD;i++)
  {
```

```
        if(i<-20)
          A=0.3*i+inicioAFD;
        else
          A=2*i-34+inicioAFD;

        if (A==0)
          A=A+0.001;
        tet1=asin((double)D1/A);
        tet2=asin((double)D2/A);
        N=(2*(tet2-tet1)-sin(2*tet2)+sin(2*tet1))*K/pi;
        N=N+(4*K*D1/(pi*A))*(cos(tet2)-cos(tet1))+(4*U/(pi*A))*cos(tet2);

        if(N==0)
          {
            graphicsStop();
            resume();
            puts("\a");
            messageBox("LOS DATOS SON INCONSISTENTES",
             mfError | mfOKButton);
            return 1;
          }

        realFD[i]=-1/N;
        imagFD[i]=0;
      }

    tipoNL=6;
    pto1=U;
    pto2=D1;
    pto3=D2;
    settextjustify(0,0);
    graphicsStop();
    resume();
    return inicioAFD;
}




 /*CASO DE UNA SATURACION CON PENDIENTE*/
int TVAesfudes::sat_Pend()
 {
    int i;
    float A, U, D, K1, K2, N;
    double tet;
    char ch;

    /*INICIALIZACION DE ARREGLOS*/
    for(i=0;i<60;i++)
      realFD[i]=0;
    for(i=0;i<60;i++)
      imagFD[i]=0;

    do
      {
       do
         {
           i=1;
           gotoxy(1,15);
           puts("Ingrese el valor de U");
           gotoxy(30,15);
           puts("U= ");
           gotoxy(33,15);
           fflush(stdin);
           while(scanf("%f",&U)==0)
             {
               fflush(stdin);
```

```
        gotoxy(33,15);
        printf("\a         ");
        gotoxy(1,16);
        puts("                                        ");
        gotoxy(33,15);
       }
    gotoxy(1,16);
    puts("                                      ");
    gotoxy(33,15);
    printf("%f",U);

    if(U<-0)
      {
       gotoxy(1,16);
       setcolor(RED);
       puts("\aU DEBE SER MAYOR QUE CERO");
       setcolor(YELLOW);
       gotoxy(33,15);
       puts("            ");
       i-0;
      }
  }while(i==0);

gotoxy(1,16);
puts("                                      ");
do
  {
    i-1;
    gotoxy(1,17);
    puts("Ingrese el valor de D");
    gotoxy(30,17);
    puts("D= ");
    fflush(stdin);
    gotoxy(33,17);
    while(scanf("%f",&D)--0)
      {
       fflush(stdin);
       gotoxy(33,17);
       puts("\a             ");
       gotoxy(1,18);
       puts("                                        ");
       gotoxy(33,17);
      }
    gotoxy(1,18);
    puts("                                      ");
    gotoxy(33,17);
    printf("%f",D);

    if(D<-0)
      {
       gotoxy(1,18);
       setcolor(RED);
       puts("\aD DEBE SER MAYOR QUE CERO");
       setcolor(YELLOW);
       gotoxy(33,17);
       puts("            ");
       i-0;
      }
  }while(i==0);

gotoxy(1,18);
puts("                                  ");

gotoxy(1,19);
puts("Ingrese el valor de K");
gotoxy(30,19);
puts("K= ");
fflush(stdin);
gotoxy(33,19);
```

```
    while(scanf("%f",&K2)--0)
      {
        fflush(stdin);
        gotoxy(33,19);
        puts("\a              ");
        gotoxy(1,20);
        puts("                                        ");
        gotoxy(33,19);
      }
    gotoxy(1,20);
    puts("                                    ");
    gotoxy(33,19);
    printf("%f",K2);

    gotoxy(1,21);
    setcolor(RED);
    printf("LOS DATOS SON CORRECTOS s/n\t");
    setcolor(YELLOW);
    fflush(stdin);
    ch=getch();
    if(isupper(ch))
    ch=tolower(ch);
    if(ch!='s')
      {
        gotoxy(1,15);
        puts("\a                                              ");
        gotoxy(1,17);
        puts("                                          ");
        gotoxy(1,19);
        puts("                                          ");
        gotoxy(1,21);
        puts("                                          ");

      }
  }while(ch!='s');

/*CALCULO DE -1/N*/
if (D==0)
  K1=U/(D+0.001);
else
  K1=U/D;
inicioAFD=D+0.001;
for(i=0;i<num_datosFD;i++)
  {
    if(i<=20)
      A=0.3*i+inicioAFD;
    else
      A=2*i-34+inicioAFD;

    if(A==0)
      A=A+0.001;
    tet=asin((double)D/A);
    N = (2*tet-sin(2*tet))*K1/pi + (pi-2*tet+sin(2*tet))*K2/pi;
    N = N -4*K2*D*cos(tet)/(pi*A) + 4*U*cos(tet)/(pi*A);

    if(N==0)
      {
        graphicsStop();
        resume();
        puts("\a");
        messageBox("LOS DATOS SON INCONSISTENTES",
         mfError | mfOKButton);
        return 1;
      }

    realFD[i]=-1/N;
    imagFD[i]=0;
  }
```

```
    tipoNL=7;
    pto1=U;
    pto2=D;
    pto3=K2;
    settextjustify(0,0);
    graphicsStop();
    resume();
    return inicioAFD;
}



 /*CASO DE UNA SATURACION CON ZONA MUERTA Y PENDIENTE*/
int TVAesfudes::sat_ZMyP()
{
    int i;
    float A, U, D1, D2, K1, K2, N;
    double tet1, tet2;
    char ch;

    /*INICIALIZACION DE ARREGLOS*/
    for(i=0;i<60;i++)
      realFD[i]=0;
    for(i=0;i<60;i++)
      imagFD[i]=0;

    do
      {
       do
         {
          i=1;
          gotoxy(1,14);
          puts("Ingrese el valor de U");
          gotoxy(30,14);
          puts("U= ");
          gotoxy(33,14);
          fflush(stdin);
          while(scanf("%f",&U)==0)
            {
             fflush(stdin);
             gotoxy(33,14);
             printf("\a          ");
             gotoxy(1,15);
             puts("                                        ");
             gotoxy(33,14);
            }
          gotoxy(1,15);
          puts("                                      ");
          gotoxy(33,14);
          printf("%f",U);

          if(U<=0)
            {
             gotoxy(1,15);
             setcolor(RED);
             puts("\aU DEBE SER MAYOR QUE CERO");
             setcolor(YELLOW);
             gotoxy(33,14);
             puts("             ");
             i=0;
            }
         }while(i--0);

       gotoxy(1,15);
       puts("                                      ");

       do
```

```
  {
    i=1;
    gotoxy(1,16);
    puts("Ingrese el valor de D1");
    gotoxy(30,16);
    puts("D1= ");
    fflush(stdin);
    gotoxy(33,16);
    while(scanf("%f",&D1)==0)
      {
       fflush(stdin);
       gotoxy(33,16);
       puts("\a              ");
       gotoxy(1,17);
       puts("                                    ");
       gotoxy(33,16);
      }
    gotoxy(1,17);
    puts("                                    ");
    gotoxy(33,16);
    printf("%f",D1);

    if(D1<=0)
      {
       gotoxy(1,17);
       setcolor(RED);
       puts("\aD1 DEBE SER MAYOR QUE CERO");
       setcolor(YELLOW);
       gotoxy(33,16);
       puts("            ");
       i=0;
      }
  }while(i--=0);

gotoxy(1,17);
puts("                                    ");

do
  {
    i=1;
    gotoxy(1,18);
    puts("Ingrese el valor de D2");
    gotoxy(30,18);
    puts("D2= ");
    gotoxy(33,18);
    fflush(stdin);
    while(scanf("%f",&D2)==0)
      {
       fflush(stdin);
       gotoxy(33,18);
       printf("\a         ");
       gotoxy(1,19);
       puts("                                    ");
       gotoxy(33,18);
      }
    gotoxy(1,19);
    puts("                                    ");
    gotoxy(33,18);
    printf("%f",D2);

    if(D2<=D1)
      {
       gotoxy(1,19);
       setcolor(RED);
       puts("\aD2 DEBE SER MAYOR QUE D1");
       setcolor(YELLOW);
       gotoxy(33,18);
       puts("            ");
       i=0;
```

```
        }
      |while(i--0);

    gotoxy(1,19);
    puts("                                          ");
    gotoxy(1,20);
    puts("Ingrese el valor de K");
    gotoxy(30,20);
    puts("K- ");
    gotoxy(33,20);
    fflush(stdin);
    while(scanf("%f",&K2)--0)
      {
        fflush(stdin);
        gotoxy(33,20);
        printf("\a         ");
        gotoxy(1,21);
        puts("                                          ");
        gotoxy(33,20);
      }
    gotoxy(1,21);
    puts("                                      ");
    gotoxy(33,20);
    printf("%f",K2);

    gotoxy(1,22);
    setcolor(RED);
    printf("LOS DATOS SON CORRECTOS s/n\t");
    setcolor(YELLOW);
    fflush(stdin);
      ch-getch();
    if(isupper(ch))
      ch-tolower(ch);
    if(ch!-'s')
      {
        gotoxy(1,14);
        puts("\a                                                ");
        gotoxy(1,16);
        puts("                                              ");
        gotoxy(1,18);
        puts("                                              ");
        gotoxy(1,20);
        puts("                                              ");
        gotoxy(1,22);
        puts("                                              ");
      }
  |while(ch!-'s');

/*CALCULO DE -1/N*/
K1-D2-D1;
if (K1--0)
  K1-K1+0.001;
K1-U/K1;
inicioAFD-D2+0.001;
for(i-0;i<num_datosFD;i++)
  {
    if(i<-20)
      A-0.3*i+inicioAFD;
    else
      A-2*i-34+inicioAFD;

    if (A==0)
      A-A+0.001;
    tet1-asin((double)D1/A);
    tet2-asin((double)D2/A);
    N - (2*tet2-2*tet1-sin(2*tet2)+sin(2*tet1))*K1/pi;
    N - N + (cos(tet2)-cos(tet1))*4*K1*D1/(pi*A);
    N - N + (pi-2*tet2+sin(2*tet2))*K2/pi;
    N - N - cos(tet2)*4*K2*D2/(pi*A) + cos(tet2)*4*U/(pi*A);
```

```
        if(N==0)
          {
            graphicsStop();
            resume();
            puts("\a");
            messageBox("LOS DATOS SON INCONSISTENTES",
             mfError | mfOKButton);
            return 1;
          }
        realFD[i]=-1/N;
        imagFD[i]=0;
      }

    tipoNL=8;
    pto1=U;
    pto2=D1;
    pto3=D2;
    pto4=K2;
    settextjustify(0,0);
    graphicsStop();
    resume();
    return inicioAFD;
}
```

```
/*-------------------------------------------------------*/
/*                                                       */
/*   PROGRAMA PARA ANALIZAR ESTABILIDAD                  */
/*   UTILIZANDO LA FUNCION DESCRIPTIVA                   */
/*     VER LAS FUNCIONES ESCOGIDAS                       */
/*    Wilmer Cárdenas Burbano 1994                       */
/*-------------------------------------------------------*/


#define Uses_MsgBox
#define Uses_TApplication

#include <tv.h>
#include <stdio.h>
#include <string.h>
#include <graphics.h>
#include <conio.h>

#include "funcion.h"

extern float numerador[];
extern float denominador[];
extern int gradnum, gradden;
extern int tipoNL;
extern float pto1, pto2, pto3, pto4;
extern float Kp, Kd, Ki;


void TVAesfudes::verFT()
{
  int i, xmax, Y;

  const MAXSIZE = 80;
  char errorMsg[MAXSIZE];

  suspend();
  if (graphicsStart() == False)
    {
      strcpy(errorMsg,grapherrormsg(graphresult()));
      strcat( errorMsg,"." );
      messageBox(errorMsg, mfError | mfOKButton);
    }
  else
    {
      setbkcolor(BLUE);
      xmax = getmaxx();
      cleardevice();
      settextjustify(1,1);
      setcolor(WHITE);
        outtextxy(xmax/2,5,"PROGRAMA SEMIGRAFICO PARA ANALIZAR ESTABILIDAD USANDO LA FUNCION
DESCRIPTIVA");
      moveto(0,0);
      setcolor(GREEN);
      gotoxy(1,2);
      puts("LA PRESENTE FUNCION DE TRANSFERENCIA ES:");
      settextjustify(0,0);
      outtextxy(160,70,"  a0+a1 +.....+an");
      outtextxy(160,90," b0+b1 +.....+bm ");
      outtextxy(160,67,"       1              n");
      outtextxy(160,87,"      1                m");
      outtextxy(120,79,"FT= ");
      setlinestyle(0,0,1);
      line(150,75,400,75);

      setcolor(YELLOW);
      gotoxy(5,8);
      printf("GRADO DEL NUMERADOR   = %d\n",gradnum);
      gotoxy(5,9);
      printf("GRADO DEL DENOMINADOR = %d\n\n",gradden);
```

```
      gotoxy(10,11);
      puts("NUMERADOR");
      Y=12;
      for(i=0;i<=gradnum;i++)
        {
          gotoxy(10,Y+i);
          printf("a%d = %9.4f\n",i,numerador[i]);
        }

      gotoxy(40,11);
      puts("DENOMINADOR");
      for(i=0;i<=gradden;i++)
        {
          gotoxy(40,Y+i);
          printf("b%d = %9.4f\n",i,denominador[i]);
        }
      gotoxy(40,25);
      setcolor(MAGENTA);
      puts("Presione una tecla para continuar");
      getch();
      graphicsStop();
    }

  resume();
}



void TVAesfudes::verGc()
{
  int xmax;

  suspend();
  graphicsStart();
  setbkcolor(BLUE);

  xmax = getmaxx();
  settextjustify(0,0);

  cleardevice();
  settextjustify(1,1);
  setcolor(WHITE);
    outtextxy(xmax/2,5,"PROGRAMA SEMIGRAFICO PARA ANALIZAR ESTABILIDAD USANDO LA FUNCION
DESCRIPTIVA");
  moveto(0,0);
  setcolor(GREEN);
  gotoxy(1,2);
  puts("COMPENSACION TIPO PID\n");
  settextjustify(0,0);
  outtextxy(150,70,"Gc = Kp(1+ Ki/s + Kd*s)");
  setcolor(YELLOW);

  gotoxy(15,10);
  printf("Kp = %f",Kp);
  gotoxy(15,12);
  printf("Kd = %f",Kd);
  gotoxy(15,14);
  printf("Ki = %f",Ki);

  gotoxy(40,25);
  setcolor(MAGENTA);
  puts("Presione una tecla para continuar");

  getch();
  graphicsStop();
  resume();
}
```

```
void TVAesfudes::verFD()
{
    if(tipoNL==1)
      {
       gotoxy(1,15);
       printf("U =   %f",pto1);
      }

    if(tipoNL==2)
      {
       gotoxy(10,15);
       printf("U =   %f",pto1);
       gotoxy(10,17);
       printf("D =   %f",pto2);
      }

    if(tipoNL==3)
      {
       gotoxy(10,15);
       printf("U =   %f",pto1);
       gotoxy(10,17);
       printf("K =   %f",pto2);
      }

    if(tipoNL==4)
      {
       gotoxy(10,15);
       printf("U =   %f",pto1);
       gotoxy(10,17);
       printf("D =   %f",pto2);
       gotoxy(10,19);
       printf("K =   %f",pto3);

      }

    if(tipoNL==5)
      {
       gotoxy(10,15);
       printf("U =   %f",pto1);
       gotoxy(10,17);
       printf("D =   %f",pto2);
      }

    if(tipoNL==6)
      {
       gotoxy(10,15);
       printf("U =   %f",pto1);
       gotoxy(10,17);
       printf("D1=   %f",pto2);
       gotoxy(10,19);
       printf("D2=   %f",pto3);
      }

    if(tipoNL==7)
      {
       gotoxy(10,15);
       printf("U =   %f",pto1);
       gotoxy(10,17);
       printf("D =   %f",pto2);
       gotoxy(10,19);
       printf("K =   %f",pto3);
      }

    if(tipoNL==8)
      {
       gotoxy(10,15);
```

```
        printf("U -   %f",pto1);
        gotoxy(10,17);
        printf("D1-   %f",pto2);
        gotoxy(10,19);
        printf("D2-   %f",pto3);
        gotoxy(10,21);
        printf("K -   %f",pto4);
      }

    if(tipoNL==9)
      {
        gotoxy(10,15);
        printf("U -   %f",pto1);
        gotoxy(10,17);
        printf("D -   %f",pto2);
      }

    if(tipoNL==10)
      {
        gotoxy(10,15);
        printf("U -   %f",pto1);
        gotoxy(10,17);
        printf("D1-   %f",pto2);
        gotoxy(10,19);
        printf("D2-   %f",pto3);
      }

    if(tipoNL==11)
      {
        gotoxy(10,15);
        printf("U -   %f",pto1);
        gotoxy(10,17);
        printf("D1-   %f",pto2);
        gotoxy(10,19);
        printf("D2-   %f",pto3);
      }

    if(tipoNL==12)
      {
        gotoxy(10,15);
        printf("U -   %f",pto1);
        gotoxy(10,17);
        printf("D1=   %f",pto2);
        gotoxy(10,19);
        printf("D2-   %f",pto3);
        gotoxy(10,21);
        printf("D3-   %f",pto4);
      }

  gotoxy(40,25);
  setcolor(MAGENTA);
  puts("Presione una tecla para continuar");
  getch();
  graphicsStop();
  resume();
}
```

```
/*-------------------------------------------------------*/
/*                                                       */
/*   PROGRAMA PARA ANALIZAR ESTABILIDAD                  */
/*   UTILIZANDO LA FUNCION DESCRIPTIVA                   */
/*                                                       */
/*               GRAFICAR                                */
/*                                                       */
/*   Wilmer Cárdenas Burbano 1994                        */
/*-------------------------------------------------------*/


#define Uses_MsgBox
#define Uses_TApplication

#define pi 3.141592

#include <tv.h>
#include <string.h>
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#include <math.h>

#include "funcion.h"

extern float realFT[];
extern float imagFT[];
extern float realFD[];
extern float imagFD[];
extern float realFTc[];
extern float imagFTc[];

extern int factor_eje;
extern int longit_vect;
extern int num_datosFD;

extern int Ptocorte;
extern float Wc, Ac1, Ac2;
extern int tipoNL;
extern float pto1, pto2, pto3, pto4;


int TVAesfudes::graficar()
{
    int i;
    const MAXSIZE = 80;
    char errorMsg[MAXSIZE];

    factor_eje=200;
    suspend();
    i=1;

    if (graphicsStart() == False)
      {
       strcpy(errorMsg,grapherrormsg(graphresult()));
       strcat( errorMsg,"." );
       messageBox(errorMsg, mfError | mfOKButton);
       i=0;
      }
    return i;
}
```

```
/*FUNCION DE TRAZADO DE EJES*/
void TVAesfudes::trazoejes()
 {
   int i, j, xmax, ymax, xo, yo;
   static char *num_graf[]={"","1","2","3","4","5","6","7","8","9","10","11","12","13"};

   cleardevice();
   xmax = getmaxx();
   ymax = getmaxy();
   xo=xmax/2;
   yo=ymax/2;

   /*TITULO*/
   settextjustify(1,1);
   outtextxy(xmax/2,5,"PROGRAMA SEMIGRAFICO PARA ANALIZAR");
   outtextxy(xmax/2,15,"ESTABILIDAD USANDO LA FUNCION DESCRIPTIVA");

   setcolor(6);
   setlinestyle(0,0,0);
   line(0,ymax/2, xmax, ymax/2);
   line(xmax/2,0,xmax/2,ymax);


   /*GRAFICO DE LOS MINOR TICKS*/
   for(i=0;i<xmax/2;i+=factor_eje/10)
    line(i+xo,yo,i+xo,yo+3);

   for(i=0;i<xmax/2;i+=factor_eje/10)
    line(-i+xo,yo,-i+xo,yo+3);

   for(i=0;i<ymax/2;i+=factor_eje/10)
    line(xo,i+yo,xo+3,i+yo);

   for(i=0;i<ymax/2;i+=factor_eje/10)
    line(xo,-i+yo,xo+3,-i+yo);

    /*GRAFICO DE LOS MEDIUMTICKS*/
    for(i=0;i<xmax/2;i+=factor_eje/2)
      line(i+xo,yo,i+xo,yo+6);

    for(i=0;i<xmax/2;i+=factor_eje/2)
      line(-i+xo,yo,-i+xo,yo+6);

    for(i=0;i<ymax/2;i+=factor_eje/2)
      line(xo,i+yo,xo+6,i+yo);

    for(i=0;i<ymax/2;i+=factor_eje/2)
      line(xo,-i+yo,xo+6,-i+yo);


    /*GRAFICO DE LOS MAYORTICKS*/
    j=0;
    for(i=0;i<xmax/2;i+=factor_eje)
      {
       line(i+xo,yo,i+xo,yo+9);
       outtextxy(i+xo,yo+20,num_graf[j]);
       j++;
      }

    j=0;
    for(i=0;i<xmax/2;i+=factor_eje)
      {
       line(-i+xo,yo,-i+xo,yo+9);
       outtextxy(-i+xo,yo+20,num_graf[j]);
       j++;
      }

    j=0;
    for(i=0;i<ymax/2;i+=factor_eje)
```

```
      {
       line(xo,i+yo,xo+9,i+yo);
       outtextxy(xo+20,i+yo,num_graf[j]);
       j++;
      }

    j=0;
    for(i=0;i<ymax/2;i+=factor_eje)
      {
       line(xo,-i+yo,xo+9,-i+yo);
       outtextxy(xo+20,-i+yo,num_graf[j]);
       j++;
      }
 }



/*FUNCION DE GRAFICACION DE LA FUNCION DE TRANSFERENCIA*/
void TVAesfudes::graficFT()
 {
    int i, Xmax, Ymax, xo, yo;
    float x1, x2, y1, y2;

    Xmax=getmaxx();
    Ymax=getmaxy();
    xo = getmaxx()/2;
    yo = getmaxy()/2;
    setcolor(getmaxcolor());
    gotoxy(5,4);
    puts("      EJE X      EJE Y");

    for (i=0;i<longit_vect-1;i++)
      {
       x1=realFT[i]*factor_eje;
       y1=imagFT[i]*factor_eje;
       x2=realFT[i+1]*factor_eje;
       y2=imagFT[i+1]*factor_eje;
       gotoxy(5,5);
       if (i<-2)
         {
          printf("FT  %9.3e   %9.3e\r", realFT[i+1], imagFT[i+1]);
          line(xo+x1, yo-y1, xo+x2, yo-y2);
         }
       else
         {
          if(xo+x2>=0 && xo+x2<=Xmax && yo-y2>=0 && yo-y2<Ymax)
            {
             printf("FT  %9.3e  %9.3e\r", realFT[i+1], imagFT[i+1]);
             line(xo+x1, yo-y1, xo+x2, yo-y2);
            }
         }
      }

    /*MENSAJE CAMBIAR DE ESCALA OPCIONAL*/
    setcolor(6);
    outtextxy(450,50,"CAMBIAR DE ESCALA");
    outtextxy(450,70,"+ Aumentar escala");
    outtextxy(450,90,"- Disminuir escala");

    /*PRESENTACION DEL PUNTO DE CORTE*/
    gotoxy(60,22);
    puts("PUNTOS DE CORTE");
    if(Ptocorte==1)
      {
       gotoxy(62,23);
       printf("Wc= %6.3f",Wc);
       if(Ac1==-1 || Ac2==-1)
         {
```

```
            gotoxy(62,24);
            if(Ac1==-1)
              printf("Ac= %6.3f",Ac2);
            else
              printf("Ac= %6.3f",Ac1);
          }
        else
          {
            gotoxy(62,24);
            printf("Ac1= %6.3f",Ac1);
            gotoxy(62,25);
            printf("Ac2= %6.3f",Ac2);
          }
      }
    else
      {
       gotoxy(65,23);
       puts("NO HAY");
      }
 }


void TVAesfudes::graficGc()
{
    int i, Xmax, Ymax, xo, yo;
    float x1, x2, y1, y2;

    Xmax=getmaxx();
    Ymax=getmaxy();
    xo = getmaxx()/2;
    yo = getmaxy()/2;
    setcolor(getmaxcolor());
    gotoxy(5,4);
    puts("      EJE X      EJE Y");

    for (i=0;i<longit_vect-1;i++)
      {
       x1=realFTc[i]*factor_eje;
       y1=imagFTc[i]*factor_eje;
       x2=realFTc[i+1]*factor_eje;
       y2=imagFTc[i+1]*factor_eje;
       gotoxy(5,5);
       if (i<-2)
         {
           printf("FTc  %9.3e   %9.3e\r", realFTc[i+1], imagFTc[i+1]);
           line(xo+x1, yo-y1, xo+x2, yo-y2);
         }
       else
         {
           if(xo+x2>=0 && xo+x2<=Xmax && yo-y2>=0 && yo-y2<Ymax)
             {
              printf("FTc  %9.3e  %9.3e\r", realFTc[i+1], imagFTc[i+1]);
              line(xo+x1, yo-y1, xo+x2, yo-y2);
             }
         }
      }

    /*MENSAJE CAMBIAR DE ESCALA OPCIONAL*/
    setcolor(6);
    outtextxy(450,50,"CAMBIAR DE ESCALA");
    outtextxy(450,70,"+ Aumentar escala");
    outtextxy(450,90,"- Disminuir escala");

    /*PRESENTACION DEL PUNTO DE CORTE*/
    gotoxy(60,22);
    puts("PUNTOS DE CORTE");

                1)
```

```
      {
       gotoxy(62,23);
       printf("Wc= %6.3f",Wc);
       if(Ac1==-1 || Ac2==-1)
         {
           gotoxy(62,24);
           if(Ac1==-1)
             printf("Ac= %6.3f",Ac2);
           else
             printf("Ac= %6.3f",Ac1);
         }
       else
         {
           gotoxy(62,24);
           printf("Ac1= %6.3f",Ac1);
           gotoxy(62,25);
           printf("Ac2= %6.3f",Ac2);
         }
      }
    else
      {
       gotoxy(65,23);
       puts("NO HAY");
      }
}



/*FUNCION DE GRAFICACION DE LA FUNCION DESCRIPTIVA*/
int TVAesfudes::graficFD()
 {
    int i, j, Xmax, Ymax, xo, yo;
    float x1, x2, y1, y2;
    char ch;

    Xmax=getmaxx();
    Ymax=getmaxy();
    xo = getmaxx()/2;
    yo = getmaxy()/2;
    setcolor(14);

    for (i=0;i<num_datosFD-1;i++)
      {
       x1=realFD[i]*factor_eje;
       y1=imagFD[i]*factor_eje;
       x2=realFD[i+1]*factor_eje;
       y2=imagFD[i+1]*factor_eje;

       if(x1>30000)
         x1=30000;
       if(x1<-30000)
         x1=-30000;
       if(x2>30000)
         x2=30000;
       if(x2<-30000)
         x2=-30000;

       if(y1>30000)
         y1=30000;
       if(y1<-30000)
         y1=-30000;
       if(y2>30000)
         y2=30000;
       if(y2<-30000)
         y2=-30000;

       gotoxy(5,6);
```

```
      if (i<-2)
        {
          printf("FD  %9.3e  %9.3e\r", realFD[i+1], imagFD[i+1]);
          line(xo+x1, yo-y1, xo+x2, yo-y2);
        }
      else
        {
          if(xo+x2>-0 && xo+x2<-Xmax && yo-y2>-0 && yo-y2<Ymax)
            {
              printf("FD  %9.3e  %9.3e\r", realFD[i+1], imagFD[i+1]);
              line(xo+x1, yo-y1, xo+x2, yo-y2);
            }
        }
    }
  setcolor(15);

  /*CAMBIAR DE ESCALA*/
  do
    {
      j-1;
      i-0;
      fflush(stdin);
      ch-getch();
      if(ch--'+' || ch--'-')
        {
          i-1;
          if(ch--'+')
            {
              factor_eje*-2;
              if(factor_eje>1600)
                {
                  gotoxy(1,1);
                  puts("\a");
                  factor_eje-1600;
                  j-0;
                }
            }
          else
            {
              factor_eje/-2;
              if (factor_eje<25)
                {
                  gotoxy(1,1);
                  puts("\a");
                  factor_eje=25;
                  j-0;
                }
            }
        }
    }while(j--0);
  return i;
}



// Graficación de la respuesta de la nolinealidad escogida ante una entrada sinusoidal.
int TVAesfudes::grafNL()
  {
    int i, xmax, ymax, sig1, sig2;
    float ang, ang2, ang3, ang4;
    setbkcolor(BLUE);
    xmax-getmaxx();
    ymax-getmaxy();
    cleardevice();
    setcolor(6);
    setlinestyle(0,0,0);
```

```
line(xmax/10,ymax/2+200,xmax/10,ymax/2-200);  /*Eje Y*/
line(xmax/10-20,ymax/2,xmax/10+600,ymax/2);   /*Eje X*/
setcolor(15);
for(i=0;i<540;i++)
  {
    line(xmax/10+i,ymax/2-100*sin(i*pi/180),
         xmax/10+i+1,ymax/2-100*sin((i+2)*pi/180));
    i++;
  }
outtextxy(xmax/10-50,ymax/2-100, "A");

setlinestyle(0,0,3);
if(tipoNL==1)
  {
    //VERTICALES
    line(xmax/10+180, ymax/2-pto1*30, xmax/10+180, ymax/2+pto1*30);
    line(xmax/10+360, ymax/2-pto1*30, xmax/10+360, ymax/2+pto1*30);

    //HORIZONTALES
    line(xmax/10, ymax/2-pto1*30, xmax/10+180, ymax/2-pto1*30);
    line(xmax/10+360, ymax/2-pto1*30, xmax/10+540, ymax/2-pto1*30);
    line(xmax/10+180, ymax/2+pto1*30, xmax/10+360, ymax/2+pto1*30);

    settextjustify(1,1);
    outtextxy(xmax/2,5,"CASO DE UN RELE IDEAL");
    outtextxy(xmax/10-50,ymax/2-pto1*30, "U");
  }

if(tipoNL==2)
  {
    if(pto2*30>100)
      ang=30;
    else
      ang=asin(pto2*30/100)*180/pi;
    //VERTICALES
    line(xmax/10+ang, ymax/2-pto1*30, xmax/10+ang, ymax/2);
    line(xmax/10+180-ang, ymax/2-pto1*30, xmax/10+180-ang, ymax/2);
    line(xmax/10+180+ang, ymax/2, xmax/10+180+ang, ymax/2+pto1*30);
    line(xmax/10+360-ang, ymax/2, xmax/10+360-ang, ymax/2+pto1*30);
    line(xmax/10+360+ang, ymax/2-pto1*30, xmax/10+360+ang, ymax/2);
    line(xmax/10+540-ang, ymax/2-pto1*30, xmax/10+540-ang, ymax/2);

    //EJE HORIZONTAL
    line(xmax/10, ymax/2, xmax/10+ang, ymax/2);
    line(xmax/10+180-ang, ymax/2, xmax/10+180+ang, ymax/2);
    line(xmax/10+360-ang, ymax/2, xmax/10+360+ang, ymax/2);
    line(xmax/10+540-ang, ymax/2, xmax/10+540, ymax/2);

    //HORIZONTALES
    line(xmax/10+ang, ymax/2-pto1*30, xmax/10+180-ang, ymax/2-pto1*30);
    line(xmax/10+360+ang, ymax/2-pto1*30, xmax/10+540-ang, ymax/2-pto1*30);
    line(xmax/10+180+ang, ymax/2+pto1*30, xmax/10+360-ang, ymax/2+pto1*30);

    settextjustify(1,1);
    outtextxy(xmax/2,5,"CASO DE UN RELE CON ZONA MUERTA");
    outtextxy(xmax/10-50,ymax/2-pto2*30, "D");
    outtextxy(xmax/10-50,ymax/2-pto1*30, "U");
  }

if(tipoNL==3)
  {
    //VERTICALES
    line(xmax/10, ymax/2-pto1*5, xmax/10, ymax/2+pto1*5);
    line(xmax/10+180, ymax/2-pto1*5, xmax/10+180, ymax/2+pto1*5);
    line(xmax/10+360, ymax/2-pto1*5, xmax/10+360, ymax/2+pto1*5);
    line(xmax/10+540, ymax/2-pto1*5, xmax/10+540, ymax/2+pto1*5);


    //CURVAS
```

```
    for(i=0;i<=540;i++)
      {
        if((i>=0 && i<=180) || i>=360)
          line(xmax/10+i,ymax/2-(100*sin(i*pi/180)*pto2+pto1*5),
               xmax/10+i+1,ymax/2-(100*sin((i+1)*pi/180)*pto2+pto1*5));
        else
          line(xmax/10+i,ymax/2-(100*sin(i*pi/180)*pto2-pto1*5),
               xmax/10+i+1,ymax/2-(100*sin((i+1)*pi/180)*pto2-pto1*5));
      }

    settextjustify(1,1);
    outtextxy(xmax/2,5,"CASO DE UN RELE CON PENDIENTE");
    outtextxy(xmax/10-50,ymax/2-pto1*5, "U");
    outtextxy(xmax/10-40,ymax/2-(100*pto2+pto1*5), "KA+U");
  }

if(tipoNL==4)
  {
    if(pto2*30>100)
      ang=30;
    else
      ang=asin(pto2*30/100)*180/pi;

    //VERTICALES
    line(xmax/10+ang, ymax/2-pto1*30, xmax/10+ang, ymax/2);
    line(xmax/10+180-ang, ymax/2-pto1*30, xmax/10+180-ang, ymax/2);
    line(xmax/10+180+ang, ymax/2, xmax/10+180+ang, ymax/2+pto1*30);
    line(xmax/10+360-ang, ymax/2, xmax/10+360-ang, ymax/2+pto1*30);
    line(xmax/10+360+ang, ymax/2-pto1*30, xmax/10+360+ang, ymax/2);
    line(xmax/10+540-ang, ymax/2-pto1*30, xmax/10+540-ang, ymax/2);

    //EJE HORIZONTAL
    line(xmax/10, ymax/2, xmax/10+ang, ymax/2);
    line(xmax/10+180-ang, ymax/2, xmax/10+180+ang, ymax/2);
    line(xmax/10+360-ang, ymax/2, xmax/10+360+ang, ymax/2);
    line(xmax/10+540-ang, ymax/2, xmax/10+540, ymax/2);


    //CURVAS
    for(i=0;i<=540;i++)
      {
        if((i>=ang && i<=(180-ang)) || (i>=(360+ang) && (i<=(540-ang))))
          line(xmax/10+i,ymax/2-((100*sin(i*pi/180)-pto2*30)*pto3+pto1*30),
               xmax/10+i+1,ymax/2-((100*sin((i+1)*pi/180)-pto2*30)*pto3+pto1*30));
        if((i>=180+ang) && (i<=360-ang))
          line(xmax/10+i,ymax/2-((100*sin(i*pi/180)+pto2*30)*pto3-pto1*30),
               xmax/10+i+1,ymax/2-((100*sin((i+1)*pi/180)+pto2*30)*pto3-pto1*30));
      }

    settextjustify(1,1);
    outtextxy(xmax/2,5,"CASO DE UN RELE CON ZONA MUERTA Y PENDIENTE");
    outtextxy(xmax/10-50,ymax/2-pto1*30, "U");
    outtextxy(xmax/10-50,ymax/2-pto2*30, "D");
    outtextxy(xmax/10-30,ymax/2-((100-pto2*30)*pto3+pto1*30), "K(A-D)+U");
  }

if(tipoNL==5)
  {
    if(pto2*30>100)
      ang=30;
    else
      ang=asin(pto2*30/100)*180/pi;

    //HORIZONTALES
    line(xmax/10+ang, ymax/2-pto1*30, xmax/10+180-ang, ymax/2-pto1*30);
    line(xmax/10+360+ang, ymax/2-pto1*30, xmax/10+540-ang, ymax/2-pto1*30);
    line(xmax/10+180+ang, ymax/2+pto1*30, xmax/10+360-ang, ymax/2+pto1*30);
```

```
    //CURVAS
    for(i=0;i<=540;i++)
      {
        if(i<=ang || (i>=(180-ang) && i<=(180+ang))
             || (i>=(360-ang) && (i<=360+ang)) || (i>=(540-ang)))
          line(xmax/10+i,ymax/2-(100*sin(i*pi/180)*pto1/pto2),
              xmax/10+i+1,ymax/2-(100*sin((i+1)*pi/180)*pto1/pto2));
      }

    settextjustify(1,1);
    outtextxy(xmax/2,5,"CASO DE UNA SATURACION IDEAL");
    outtextxy(xmax/10-50,ymax/2-pto1*30, "U");
    outtextxy(xmax/10-50,ymax/2-pto2*30, "D");
  }

if(tipoNL==6)
  {
    if(pto2*30>100)
      ang=10;
    else
      ang=asin(pto2*30/100)*180/pi;

    if(pto3*30>100)
      {
        ang =10;
        ang2=30;
      }
    else
      ang2=asin(pto3*30/100)*180/pi;


    //HORIZONTALES
    line(xmax/10+ang2, ymax/2-pto1*30, xmax/10+180-ang2, ymax/2-pto1*30);
    line(xmax/10+360+ang2, ymax/2-pto1*30, xmax/10+540-ang2, ymax/2-pto1*30);
    line(xmax/10+180+ang2, ymax/2+pto1*30, xmax/10+360-ang2, ymax/2+pto1*30);

    //EJE HORIZONTAL
    line(xmax/10, ymax/2, xmax/10+ang, ymax/2);
    line(xmax/10+180-ang, ymax/2, xmax/10+180+ang, ymax/2);
    line(xmax/10+360-ang, ymax/2, xmax/10+360+ang, ymax/2);
    line(xmax/10+540-ang, ymax/2, xmax/10+540, ymax/2);

    //CURVAS
    for(i=0;i<=540;i++)
      {
        if((i<=ang2 && i>=ang) || (i>=(180-ang2) && i<=(180-ang))
            || (i>=(360+ang) && i<=(360+ang2))
            || (i>=(540-ang2) && i<=(540-ang)))
          line(xmax/10+i,ymax/2-(100*sin(i*pi/180)-pto2*30)*pto1/(pto3-pto2),
              xmax/10+i+1,ymax/2-(100*sin((i+1)*pi/180)-pto2*30)*pto1/(pto3-pto2));

        if((i>=(180+ang) && i<=(180+ang2)) || (i>=(360-ang2) && i<=(360-ang)))
          line(xmax/10+i,ymax/2-(100*sin(i*pi/180)+pto2*30)*pto1/(pto3-pto2),
              xmax/10+i+1,ymax/2-(100*sin((i+1)*pi/180)+pto2*30)*pto1/(pto3-pto2));
      }

    settextjustify(1,1);
    outtextxy(xmax/2,5,"CASO DE UNA SATURACION IDEAL");
    outtextxy(xmax/10-50,ymax/2-pto1*30, "U");
    outtextxy(xmax/10-50,ymax/2-pto2*30, "D1");
    outtextxy(xmax/10-50,ymax/2-pto3*30, "D2");
  }

if(tipoNL==7)
  {
    if(pto2*30>100)
      ang=30;
    else
      ang=asin(pto2*30/100)*180/pi;
```

```
    //CURVAS
    for(i=0;i<=540;i++)
      {

        if((i>=ang && i<=(180-ang)) || (i>=(360+ang) && i<=(540-ang)))
          line(xmax/10+i,ymax/2-((100*sin(i*pi/180)-pto2*30)*pto3+pto1*30),
               xmax/10+i+1,ymax/2-((100*sin((i+1)*pi/180)-pto2*30)*pto3+pto1*30));

        if((i>=(180+ang) && i<=(360-ang)))
          line(xmax/10+i,ymax/2-((100*sin(i*pi/180)+pto2*30)*pto3-pto1*30),
               xmax/10+i+1,ymax/2-((100*sin((i+1)*pi/180)+pto2*30)*pto3-pto1*30));

        if(i<=ang || (i>=(180-ang) && i<=(180+ang))
           || (i>=(360-ang) && i<=(360+ang)) || (i>=(540-ang)))
          line(xmax/10+i,ymax/2-100*sin(i*pi/180)*pto1/pto2,
               xmax/10+i+1,ymax/2-100*sin((i+1)*pi/180)*pto1/pto2);
      }

    settextjustify(1,1);
    outtextxy(xmax/2,5,"CASO DE UNA SATURACION CON PENDIENTE");
    outtextxy(xmax/10-50,ymax/2-pto1*30, "U");
    outtextxy(xmax/10-50,ymax/2-pto2*30, "D");
    outtextxy(xmax/10-30,ymax/2-((100-pto2*30)*pto3+pto1*30), "K(A-D)+U");
  }

if(tipoNL==8)
  {
    if(pto2*30>100)
      ang=10;
    else
      ang=asin(pto2*30/100)*180/pi;

    if(pto3*30>100)
      {
        ang =10;
        ang2=30;
      }
    else
      ang2=asin(pto3*30/100)*180/pi;

    //EJE HORIZONTAL
    line(xmax/10, ymax/2, xmax/10+ang, ymax/2);
    line(xmax/10+180-ang, ymax/2, xmax/10+180+ang, ymax/2);
    line(xmax/10+360-ang, ymax/2, xmax/10+360+ang, ymax/2);
    line(xmax/10+540-ang, ymax/2, xmax/10+540, ymax/2);


    //CURVAS
    for(i=0;i<=540;i++)
      {

        if((i>=ang2 && i<=(180-ang2)) || (i>=(360+ang2) && i<=(540-ang2)))
          line(xmax/10+i,ymax/2-((100*sin(i*pi/180)-pto3*30)*pto4+pto1*30),
               xmax/10+i+1,ymax/2-((100*sin((i+1)*pi/180)-pto3*30)*pto4+pto1*30));

        if((i>=(180+ang2) && i<=(360-ang2)))
          line(xmax/10+i,ymax/2-((100*sin(i*pi/180)+pto3*30)*pto4-pto1*30),
               xmax/10+i+1,ymax/2-((100*sin((i+1)*pi/180)+pto3*30)*pto4-pto1*30));

        if((i>=ang && i<=ang2) || (i>=(180-ang2) && i<=(180-ang))
           || (i>=(360+ang) && i<=(360+ang2)) || (i>=(540-ang2) && i<=(540-ang)))
          line(xmax/10+i,ymax/2-(100*sin(i*pi/180)-pto2*30)*pto1/(pto3-pto2),
               xmax/10+i+1,ymax/2-(100*sin((i+1)*pi/180)-pto2*30)*pto1/(pto3-pto2));

        if((i>=(180+ang) && i<=(180+ang2)) || (i>=(360-ang2) && i<=(360-ang)))
          line(xmax/10+i,ymax/2-(100*sin(i*pi/180)+pto2*30)*pto1/(pto3-pto2),
               xmax/10+i+1,ymax/2-(100*sin((i+1)*pi/180)+pto2*30)*pto1/(pto3-pto2));
      }
```

```
    settextjustify(1,1);
    outtextxy(xmax/2,5,"CASO DE UNA SATURACION CON ZONA MUERTA Y PENDIENTE");
    outtextxy(xmax/10-50,ymax/2-pto1*30, "U");
    outtextxy(xmax/10-50,ymax/2-pto2*30, "D1");
    outtextxy(xmax/10-50,ymax/2-pto3*30, "D2");
    outtextxy(xmax/10-25,ymax/2-((100-pto3*30)*pto4+pto1*30), "K(A-D2)+U");
   }

 if(tipoNL==9)
   {
    if(pto2*30>100)
      ang=30;
    else
      ang=asin(pto2*30/100)*180/pi;

    //VERTICALES
    line(xmax/10+ang, ymax/2-pto1*30, xmax/10+ang, ymax/2+pto1*30);
    line(xmax/10+180+ang, ymax/2-pto1*30, xmax/10+180+ang, ymax/2+pto1*30);
    line(xmax/10+360+ang, ymax/2-pto1*30, xmax/10+360+ang, ymax/2+pto1*30);

    //HORIZONTALES
    line(xmax/10+ang, ymax/2-pto1*30, xmax/10+180+ang, ymax/2-pto1*30);
    line(xmax/10+360+ang, ymax/2-pto1*30, xmax/10+540, ymax/2-pto1*30);
    line(xmax/10, ymax/2+pto1*30, xmax/10+ang, ymax/2+pto1*30);
    line(xmax/10+180+ang, ymax/2+pto1*30, xmax/10+360+ang, ymax/2+pto1*30);

    settextjustify(1,1);
    outtextxy(xmax/2,5,"CASO DE UNA HISTERESIS IDEAL");
    outtextxy(xmax/10-50,ymax/2-pto1*30, "U");
    outtextxy(xmax/10-50,ymax/2-pto2*30, "D");
   }

 if(tipoNL==10)
   {
    if(pto2*30>100)
      ang=10;
    else
      ang=asin(pto2*30/100)*180/pi;

    if(pto3*30>100)
      {
        ang =10;
        ang2=30;
      }
    else
      ang2=asin(pto3*30/100)*180/pi;

    //VERTICALES
    line(xmax/10+ang2, ymax/2-pto1*30, xmax/10+ang2, ymax/2);
    line(xmax/10+180-ang, ymax/2-pto1*30, xmax/10+180-ang, ymax/2);
    line(xmax/10+360+ang2, ymax/2-pto1*30, xmax/10+360+ang2, ymax/2);
    line(xmax/10+540-ang, ymax/2-pto1*30, xmax/10+540-ang, ymax/2);
    line(xmax/10+180+ang2, ymax/2, xmax/10+180+ang2, ymax/2+pto1*30);
    line(xmax/10+360-ang, ymax/2, xmax/10+360-ang, ymax/2+pto1*30);


    //HORIZONTALES
    line(xmax/10+ang2, ymax/2-pto1*30, xmax/10+180-ang, ymax/2-pto1*30);
    line(xmax/10+360+ang2, ymax/2-pto1*30, xmax/10+540-ang, ymax/2-pto1*30);
    line(xmax/10+180+ang2, ymax/2+pto1*30, xmax/10+360-ang, ymax/2+pto1*30);

    //EJE HORIZONTAL
    line(xmax/10, ymax/2, xmax/10+ang2, ymax/2);
    line(xmax/10+180-ang, ymax/2, xmax/10+180+ang2, ymax/2);
    line(xmax/10+360-ang, ymax/2, xmax/10+360+ang2, ymax/2);
    line(xmax/10+540-ang, ymax/2, xmax/10+540, ymax/2);
```

```
      settextjustify(1,1);
      outtextxy(xmax/2,5,"CASO DE UNA HISTERESIS CON ZONA MUERTA");
      outtextxy(xmax/10-50,ymax/2-pto1*30, "U");
      outtextxy(xmax/10-50,ymax/2-pto2*30, "D1");
      outtextxy(xmax/10-50,ymax/2-pto3*30, "D2");
    }

  if(tipoNL==11)
    {
      if(pto2*30>100)
        ang=10;
      else
        ang=asin(pto2*30/100)*180/pi;

      if(pto3*30>100)
        {
          ang =10;
          ang2=30;
        }
      else
        ang2=asin(pto3*30/100)*180/pi;


      //HORIZONTALES
      line(xmax/10+ang2, ymax/2-pto1*30, xmax/10+180-ang, ymax/2-pto1*30);
      line(xmax/10+360+ang2, ymax/2-pto1*30, xmax/10+540-ang, ymax/2-pto1*30);
      line(xmax/10+180+ang2, ymax/2+pto1*30, xmax/10+360-ang, ymax/2+pto1*30);

      //CURVAS
      for(i=0;i<=540;i++)
        {

          if((i>=0 && i<=ang2) || (i>=(360-ang) && i<=(360+ang2)))
            line(xmax/10+i,ymax/2-((100*sin(i*pi/180)-pto3*30)*2*pto1/(pto3+pto2)+pto1*30),

xmax/10+i+1,ymax/2-((100*sin((i+1)*pi/180)-pto3*30)*2*pto1/(pto3+pto2)+pto1*30));

          if((i>=(180-ang) && i<=(180+ang2)) || i>=(540-ang))
            line(xmax/10+i,ymax/2-((100*sin(i*pi/180)+pto3*30)*2*pto1/(pto3+pto2)-pto1*30),

xmax/10+i+1,ymax/2-((100*sin((i+1)*pi/180)+pto3*30)*2*pto1/(pto3+pto2)-pto1*30));
        }



      settextjustify(1,1);
      outtextxy(xmax/2,5,"CASO DE UNA HISTERESIS CON PENDIENTE");
      outtextxy(xmax/10-50,ymax/2-pto1*30, "U");
      outtextxy(xmax/10-50,ymax/2-pto2*30, "D1");
      outtextxy(xmax/10-50,ymax/2-pto3*30, "D2");

    }

  if(tipoNL==12)
    {
      if(pto2*30>100)
        ang=30;
      else
        ang=asin(pto2*30/100)*180/pi;

      if(pto3*30>100)
        ang2=30;
      else
        ang2=asin(pto3*30/100)*180/pi;

      if(pto4*30>100)
        ang3 = 30;
      else
```

```
        if((pto4+pto3-pto2)*30>100)
          {
             puts("\a");
             return 0;
          }
        else
          ang4=asin((pto4+pto3-pto2)*30/100)*180/pi;


        //HORIZONTALES
        line(xmax/10+ang4, ymax/2-pto1*30, xmax/10+180-ang2, ymax/2-pto1*30);
        line(xmax/10+360+ang4, ymax/2-pto1*30, xmax/10+540-ang2, ymax/2-pto1*30);
        line(xmax/10+180+ang4, ymax/2+pto1*30, xmax/10+360-ang2, ymax/2+pto1*30);

        //EJE HORIZONTAL
        line(xmax/10, ymax/2, xmax/10+ang3, ymax/2);
        line(xmax/10+180-ang, ymax/2, xmax/10+180+ang3, ymax/2);
        line(xmax/10+360-ang, ymax/2, xmax/10+360+ang3, ymax/2);
        line(xmax/10+540-ang, ymax/2, xmax/10+540, ymax/2);

        //CURVAS
        for(i=0;i<=540;i++)
          {
            if((i>=ang3 && i<=ang4) || (i>=(360+ang3) && i<=(360+ang4)))
              line(xmax/10+i,ymax/2-(100*sin(i*pi/180)-pto4*30)*pto1/(pto3-pto2),
                   xmax/10+i+1,ymax/2-(100*sin((i+1)*pi/180)-pto4*30)*pto1/(pto3-pto2));

            if((i>=(180-ang2) && i<=(180-ang)) || (i>=(540-ang2) && i<=(540-ang)))
              line(xmax/10+i,ymax/2-(100*sin(i*pi/180)-pto2*30)*pto1/(pto3-pto2),
                   xmax/10+i+1,ymax/2-(100*sin((i+1)*pi/180)-pto2*30)*pto1/(pto3-pto2));

            if(i>=(180+ang3) && i<=(180+ang4))
              line(xmax/10+i,ymax/2-(100*sin(i*pi/180)+pto4*30)*pto1/(pto3-pto2),
                   xmax/10+i+1,ymax/2-(100*sin((i+1)*pi/180)+pto4*30)*pto1/(pto3-pto2));

            if(i>=(360-ang2) && i<=(360-ang))
              line(xmax/10+i,ymax/2-(100*sin(i*pi/180)+pto2*30)*pto1/(pto3-pto2),
                   xmax/10+i+1,ymax/2-(100*sin((i+1)*pi/180)+pto2*30)*pto1/(pto3-pto2));

          }


        settextjustify(1,1);
        outtextxy(xmax/2,5,"CASO DE UNA HISTERESIS CON ZONA MUERTA Y PENDIENTE");
        outtextxy(xmax/10-50,ymax/2-pto1*30, "U");
        outtextxy(xmax/10-50,ymax/2-pto2*30, "D1");
        outtextxy(xmax/10-50,ymax/2-pto3*30, "D2");
        outtextxy(xmax/10-50,ymax/2-pto4*30, "D3");
      }



    setcolor(YELLOW);
    return 0;
}
```

```
/*-------------------------------------------------------*/
/*                                                       */
/*   PROGRAMA PARA ANALIZAR ESTABILIDAD                  */
/*   UTILIZANDO LA FUNCION DESCRIPTIVA                   */
/*                                                       */
/*  DETERMINACION DE LOS PUNTOS DE CRUCE                 */
/*                                                       */
/*   Wilmer Cárdenas Burbano 1994                        */
/*-------------------------------------------------------*/

#define Uses_TApplication

#define pi 3.141592

#include <tv.h>
#include <math.h>

#include "funcion.h"

extern int gradnum, gradden;
extern float numerador[];
extern float denominador[];
extern float realFD[];
extern int longit_vect;
extern float factor_dato;
extern int tipoNL;
extern float Wc, Ac1, Ac2;
extern int Ptocorte;
extern float inicioAFD;
extern float pto1, pto2, pto3, pto4;
extern float Kp, Kd, Ki;



void TVAesfudes::puntosdecruce()
{
    int i, j, ptocFT;
    float w1, w2, w3, inicio, imagFT1, imagFT2, imagFT3;
    float realFTc;
    float A1, A2, A3, realFD1, realFD2, realFD3;
    float fWA, gWA, fW, fA, gW, gA;
    float realFT1, realFT2, realFT3;
    float imagFD1, imagFD2, imagFD3;
    float denom, minFD;
    float U, D;

    //inicio es el valor inicial de frecuencia w
    inicio=factor_dato/(float)longit_vect;

    ptocFT=0;          // Vector que indica si se encontró o no
                       // pto de cruce de G(jw) con el eje real
    Ptocorte=0;
    Ac1=-1;
    Ac2=-1;

    //Análisis para determinar si G(jw) cruza el eje real
    if (tipoNL<=9)
      {
       if(tipoNL==9)
         {
           U=pto1;
           D=pto2;
         }
       w1=inicio;
       imagFT1=calc_imagFT(w1);
       if(tipoNL==9)
         imagFT1=imagFT1+pi*D/(4*U);
       w2=inicio*1000;
       imagFT2=calc_imagFT(w2);
```

```
if(tipoNL==9)
  imagFT2=imagFT2+pi*D/(4*U);

//En este caso la respuesta G(jw) cruza el eje real
if(imagFT2*imagFT1<0)
  {
    do
      {
       w3=(w1+w2)/2;
       imagFT3=calc_imagFT(w3);
       if(tipoNL==9)
         imagFT3=imagFT3+pi*D/(4*U);

       if(imagFT2*imagFT3<=0)
         {
           w1=w3;
           imagFT1=imagFT3;
         }
       else
         {
           if(imagFT1*imagFT3<=0)
             {
              w2=w3;
              imagFT2=imagFT3;
             }
         }
      }while(fabs(w1-w2)>0.0001);
    ptocFT=1;

    Wc=w1;              //Wc es la frecuancia de corte
  }

if(ptocFT==1)
  {
    // Cálculo del punto minimo o máximo de la función descriptiva
    minFD=realFD[0];
    for(i=1; i<60; i++)
      {
       if(minFD<realFD[i])
         minFD=realFD[i];
      }

    if(minFD==realFD[0])       //se buscará el máximo
      {
       for(i=1; i<60; i++)
         {
           if(minFD>realFD[i])
             minFD=realFD[i];
         }
      }

    realFTc=calc_realFT(Wc);  //Es el valor al que A debe llegar

    //calculo del primer pto de cruce
    A1=inicioAFD;
    realFD1=calc_realFD(A1)-realFTc;
    for(i=0; i<60; i++)
      {
       if(realFD[i]==minFD)
         {
           if(i<=20)
             A2=0.3*i+inicioAFD;
           else
             A2=2*i-34+inicioAFD;
         }
      }
    realFD2=minFD-realFTc;

    if(realFD2*realFD1<0)
```

```
        {
          do
            {
              A3=(A1+A2)/2;
              realFD3=calc_realFD(A3)-realFTc;
              if(realFD2*realFD3<=0)
                {
                  A1=A3;
                  realFD1=realFD3;
                }
              else
                {
                  if(realFD1*realFD3<=0)
                    {
                      A2=A3;
                      realFD2=realFD3;
                    }
                }
            }while(fabs(A1-A2)>0.0001);
          Ptocorte=1;
          Ac1=A1;
        }


      //calculo del segundo pto de cruce
      for(i=0; i<60; i++)
        {
          if(realFD[i]==minFD)
            {
              if(i<=20)
                A1=0.3*i+inicioAFD;
              else
                A1=2*i-34+inicioAFD;
            }
        }
      realFD1=minFD-realFTc;
      A2=A1+1000;
      realFD2=calc_realFD(A2)-realFTc;

      if(realFD2*realFD1<0)
        {
          do
            {
              A3=(A1+A2)/2;
              realFD3=calc_realFD(A3)-realFTc;
              if(realFD2*realFD3<=0)
                {
                  A1=A3;
                  realFD1=realFD3;
                }
              else
                {
                  if(realFD1*realFD3<=0)
                    {
                      A2=A3;
                      realFD2=realFD3;
                    }
                }
            }while(fabs(A1-A2)>0.0001);
          Ptocorte=1;
          Ac2=A1;
        }
    }
  }
else
  {
    w1=inicio+0.001;
    A1=inicioAFD+0.002;
    for(i=1;i<30;i++)
```

```
        {
          realFT1=calc_realFT(w1);
          imagFT1=calc_imagFT(w1);

          if(A1>inicioAFD)
            {
              realFD1=calc_realFD(A1);
              imagFD1=calc_imagFD(A1);
            }
          fWA=realFT1-realFD1;
          gWA=imagFT1-imagFD1;

          realFT2=calc_realFT(w1-0.001);
          realFT3=calc_realFT(w1+0.001);
          fW=(realFT3-realFT2)/0.002;

          imagFT2=calc_imagFT(w1-0.001);
          imagFT3=calc_imagFT(w1+0.001);
          gW=(imagFT3-imagFT2)/0.002;

          if(A1>inicioAFD)
            {
              realFD2=-calc_realFD(A1-0.001);
              realFD3=-calc_realFD(A1+0.001);
              fA=(realFD3-realFD2)/0.002;

              imagFD2=-calc_imagFD(A1-0.001);
              imagFD3=-calc_imagFD(A1+0.001);
              gA=(imagFD3-imagFD2)/0.002;
            }

          denom = gA*fW-gW*fA;
          w2 = w1-(gA*fWA-fA*gWA)/denom;
          A2 = A2-(fW*gWA-gW*fWA)/denom;


          if(fabs(w2-w1)<0.0001 && fabs(A2-A1)<0.0001)
            {
              Ptocorte=1;
              Wc=w2;
              Ac1=A2;
              i=30;
            }

          if(fabs(w2)>5000 || fabs(A2)>5000)
            i=30;

          w1=w2;
          A1=A2;
        }
      }
}



float TVAesfudes::calc_realFT(float w)
{
    int i;
    float  inicio, denom, realFT;
    float realnumerad, imagnumerad, realdenom, imagdenom;

        realnumerad=0;
        for(i=0;i<=gradnum;i+=2)
           realnumerad=realnumerad+pow(-1,i/2)*numerador[i]*pow(w,i);

        imagnumerad=0;
        for(i=1;i<=gradnum;i+=2)
           imagnumerad=imagnumerad+pow(-1,(i-1)/2)*numerador[i]*pow(w,i);
```

```
      realdenom=0;
      for(i=0;i<=gradden;i+=2)
         realdenom=realdenom+pow(-1,i/2)*denominador[i]*pow(w,i);

      imagdenom=0;
      for(i=1;i<=gradden;i+=2)
         imagdenom=imagdenom+pow(-1,(i-1)/2)*denominador[i]*pow(w,i);

      denom = pow(realdenom,2)+pow(imagdenom,2);

      if (denom!=0)
        {
          realFT =(realnumerad*realdenom+imagnumerad*imagdenom)/denom;
        }
  return realFT;
 }



float TVAesfudes::calc_imagFT(float w)
{
    int i;
    float  inicio, denom, imagFT;
    float realnumerad, imagnumerad, realdenom, imagdenom;

       realnumerad=0;
       for(i=0;i<=gradnum;i+=2)
          realnumerad=realnumerad+pow(-1,i/2)*numerador[i]*pow(w,i);

       imagnumerad=0;
       for(i=1;i<=gradnum;i+=2)
          imagnumerad=imagnumerad+pow(-1,(i-1)/2)*numerador[i]*pow(w,i);

       realdenom=0;
       for(i=0;i<=gradden;i+=2)
          realdenom=realdenom+pow(-1,i/2)*denominador[i]*pow(w,i);

       imagdenom=0;
       for(i=1;i<=gradden;i+=2)
          imagdenom=imagdenom+pow(-1,(i-1)/2)*denominador[i]*pow(w,i);

       denom = pow(realdenom,2)+pow(imagdenom,2);

       if (denom!=0)
         {
          imagFT =(imagnumerad*realdenom-realnumerad*imagdenom)/denom;
         }
  return imagFT;
 }



float TVAesfudes::calc_realFD(float A)
{
    int i;
    float U, D, K, K1, K2, D1, D2, D3, D4, N, NR, NI, denom;
    float realFD;
    double tet, tet1, tet2, tet3, tet4;

    if(tipoNL==1)
      {
       U=pto1;
       realFD=-pi*A/(4*U);
      }
```

```
if(tipoNL==2)
  {
   U=pto1;
   D=pto2;
   if(A==D)
     A=D+0.01;
   tet=asin((double)D/(double)A);
   realFD=-pi*A/(4*U*cos(tet));
  }

if(tipoNL==3)
  {
   U=pto1;
   K=pto2;
   realFD=-pi*(A)/(4*U+K*pi*A);
  }

if(tipoNL==4)
  {
   U=pto1;
   D=pto2;
   K=pto3;
   if (A==0)
     A=A+0.001;
   tet=asin((double)D/(double)A);
   N=(pi-2*tet+sin(2*tet))*K/pi + (4/(pi*A))*(U-K*D)*cos(tet);
   if(N!=0)
     {
       realFD=-1/N;
     }
  }

if(tipoNL==5)
  {
   U=pto1;
   D=pto2;
   if(D==0)
     K=U/(D+0.001);
   else
     K=U/D;
   if(A==0)
     A=A+0.001;
   tet=asin((double)D/A);
   N=(2*tet-sin(2*tet))*K/pi+4U*cos(tet)/(pi*A);
   if(N!=0)
     {
       realFD=-1/N;
     }
  }

if(tipoNL==6)
  {
   U=pto1;
   D1=pto2;
   D2=pto3;
   K=D2-D1;
   if(K==0)
     K=K+0.001;
   K=U/K;
   if (A==0)
     A=A+0.001;
   tet1=asin((double)D1/A);
   tet2=asin((double)D2/A);
   N=(2*(tet2-tet1)-sin(2*tet2)+sin(2*tet1))*K/pi;
   N=N+(4*K*D1/(pi*A))*(cos(tet2)-cos(tet1))+(4*U/(pi*A))*cos(tet2);
   if(N!=0)
     {
       realFD=-1/N;
     }
```

```
  }

if(tipoNL==7)
  {
    U=pto1;
    D=pto2;
    K2=pto3;
    if (D==0)
      K1=U/(D+0.001);
    else
      K1=U/D;
    if(A==0)
      A=A+0.001;
    tet=asin((double)D/A);
    N=(2*tet-sin(2*tet))*K1/pi + (pi-2*tet+sin(2*tet))*K2/pi;
    N=N -4*K2*D*cos(tet)/(pi*A) + 4*U*cos(tet)/(pi*A);
    if(N!=0)
      {
        realFD=-1/N;
      }
  }

if(tipoNL==8)
  {
    U=pto1;
    D1=pto2;
    D2=pto3;
    K2=pto4;
    K1=D2-D1;
    if (K1==0)
      K1=K1+0.001;
    K1=U/K1;
    if (A==0)
      A=A+0.001;
    tet1=asin((double)D1/A);
    tet2=asin((double)D2/A);
    N = (2*tet2-2*tet1-sin(2*tet2)+sin(2*tet1))*K1/pi;
    N = N + (cos(tet2)-cos(tet1))*4*K1*D1/(pi*A);
    N = N + (pi-2*tet2+sin(2*tet2))*K2/pi;
    N = N - cos(tet2)*4*K2*D2/(pi*A) + cos(tet2)*4*U/(pi*A);
    if(N!=0)
      {
        realFD=-1/N;
      }
  }

if(tipoNL==9)
  {
    U=pto1;
    D=pto2;
    if(A==0)
      A=A+0.001;
    tet=asin((double)D/A);
    NR = cos(tet)*4*U/(pi*A);
    NI = -sin(tet)*4*U/(pi*A);

    denom=NR*NR+NI*NI;

    if(denom!=0)
      {
        realFD=-NR/denom;
      }
  }

if(tipoNL==10)
  {
    U=pto1;
    D1=pto2;
    D2=pto3;
```

```
    if (A==0)
      A=A+0.001;
    tet1=asin((double)D1/A);
    tet2=asin((double)D2/A);
    NR = (cos(tet1)+cos(tet2))*2*U/(pi*A);
    NI = (sin(tet1)-sin(tet2))*2*U/(pi*A);

    denom=NR*NR+NI*NI;

    if(denom!=0)
      {
        realFD=-NR/denom;
      }
  }

if(tipoNL==11)
  {
    U=pto1;
    D1=pto2;
    D2=pto3;
    K=2*U/(D2+D1);
    if(A==0)
      A=A+0.001;
    tet1=asin((double)D1/A);
    tet2=asin((double)D2/A);

    NR = (2*tet1+2*tet2-sin(2*tet1)-sin(2*tet2))*K/(2*pi);
    NR = NR + (U-K*D2)*(cos(tet1)-cos(tet2))*2/(pi*A);
    NR = NR + (cos(tet2)+cos(tet1))*2U/(pi*A);

    NI = (cos(2*tet1)-cos(2*tet2))*K/(2*pi);
    NI = NI + (U-K*D2)*(sin(tet2)+sin(tet1))*2/(pi*A);
    NI = NI + (sin(tet1)-sin(tet2))*2U/(pi*A);

    denom=NR*NR+NI*NI;

    if(denom!=0)
      {
        realFD=-NR/denom;
      }
  }

if(tipoNL==12)
  {
    U=pto1;
    D1=pto2;
    D2=pto3;
    D3=pto3;
    K=(D2-D1);
    if (K==0)
      K=K+0.001;
    K=U/K;
    D4=D3+(D2-D1);
    if(D4==0)
      A=A+0.001;
    tet1=asin((double)D1/A);
    tet2=asin((double)D2/A);
    tet3=asin((double)D3/A);
    tet4=asin((double)D4/A);

    NR = (2*(tet2-tet1+tet4-tet3))*K/(2*pi);
    NR = NR + (sin(2*tet1)-sin(2*tet2)+sin(2*tet3)-sin(2*tet4))*K/(2*pi);
    NR = NR + (cos(tet4)-cos(tet3))*2*K*D3/(pi*A);
    NR = NR + (cos(tet2)-cos(tet1))*2*K*D1/(pi*A);
    NR = NR + (cos(tet4)+cos(tet2))*2*U/(pi*A);

    NI = (cos(2*tet2)-cos(2*tet1)+cos(2*tet3)-cos(2*tet4))*K/(2*pi);
    NI = NI + (sin(tet4)-sin(tet3))*2*K*D3/(pi*A);
    NI = NI + (sin(tet1)-sin(tet2))*2*K*D1/(pi*A);
```

```
        NI = NI + (sin(tet2)-sin(tet4))*2*U/(pi*A);

        denom=NR*NR+NI*NI;

        if(denom!=0)
          {
            realFD=-NR/denom;
          }
      }


    return realFD;
}



float TVAesfudes::calc_imagFD(float A)
{

    int i;
    float U, D, K, K1, K2, D1, D2, D3, D4, NR, NI, denom;
    float imagFD;
    double tet, tet1, tet2, tet3, tet4;

    if(tipoNL==10)
      {
        U=pto1;
        D1=pto2;
        D2=pto3;
        if (A==0)
          A=A+0.001;
        tet1=asin((double)D1/A);
        tet2=asin((double)D2/A);
        NR = (cos(tet1)+cos(tet2))*2*U/(pi*A);
        NI = (sin(tet1)-sin(tet2))*2*U/(pi*A);

        denom=NR*NR+NI*NI;

        if(denom!=0)
          {
            imagFD=NI/denom;
          }
      }

    if(tipoNL==11)
      {
        U=pto1;
        D1=pto2;
        D2=pto3;
        K=2*U/(D2+D1);
        if(A==0)
          A=A+0.001;
        tet1=asin((double)D1/A);
        tet2=asin((double)D2/A);

        NR = (2*tet1+2*tet2-sin(2*tet1)-sin(2*tet2))*K/(2*pi);
        NR = NR + (U-K*D2)*(cos(tet1)-cos(tet2))*2/(pi*A);
        NR = NR + (cos(tet2)+cos(tet1))*2U/(pi*A);

        NI = (cos(2*tet1)-cos(2*tet2))*K/(2*pi);
        NI = NI + (U-K*D2)*(sin(tet2)+sin(tet1))*2/(pi*A);
        NI = NI + (sin(tet1)-sin(tet2))*2U/(pi*A);

        denom=NR*NR+NI*NI;

        if(denom!=0)
          {
            imagFD=NI/denom;
```

```
        }
      }

    if(tipoNL==12)
      {
        U=pto1;
        D1=pto2;
        D2=pto3;
        D3=pto3;
      . K=(D2-D1);
        if (K==0)
          K=K+0.001;
        K=U/K;
        D4=D3+(D2-D1);
        if(D4==0)
          A=A+0.001;
        tet1=asin((double)D1/A);
        tet2=asin((double)D2/A);
        tet3=asin((double)D3/A);
        tet4=asin((double)D4/A);

        NR = (2*(tet2-tet1+tet4-tet3))*K/(2*pi);
        NR = NR + (sin(2*tet1)-sin(2*tet2)+sin(2*tet3)-sin(2*tet4))*K/(2*pi);
        NR = NR + (cos(tet4)-cos(tet3))*2*K*D3/(pi*A);
        NR = NR + (cos(tet2)-cos(tet1))*2*K*D1/(pi*A);
        NR = NR + (cos(tet4)+cos(tet2))*2*U/(pi*A);

        NI = (cos(2*tet2)-cos(2*tet1)+cos(2*tet3)-cos(2*tet4))*K/(2*pi);
        NI = NI - (sin(tet4)-sin(tet3))*2*K*D3/(pi*A);
        NI = NI - (sin(tet1)-sin(tet2))*2*K*D1/(pi*A);
        NI = NI + (sin(tet2)-sin(tet4))*2*U/(pi*A);

        denom=NR*NR+NI*NI;

        if(denom!=0)
          {
            imagFD=NI/denom;
          }
      }
    return imagFD;
}



//C O M P E N S A C I O N
void TVAesfudes::puntosdecruceGc()
{
    int i, j, ptocFT;
    float w1, w2, w3, inicio, imagFT1, imagFT2, imagFT3;
    float realFTc, imagFTc;
    float A1, A2, A3, realFD1, realFD2, realFD3;
    float fWA, gWA, fW, fA, gW, gA;
    float realFT1, realFT2, realFT3;
    float imagFD1, imagFD2, imagFD3;
    float denom, minFD;
    float U, D;

    inicio=factor_dato/(float)longit_vect;

    ptocFT=0;
    Ptocorte=0;
    if (tipoNL<=9)
      {
        if(tipoNL==9)
          {
            U=pto1;
            D=pto2;
          }
```

```
w1=inicio;
imagFT1=calc_imagFT(w1);

//Obtención de imagFT1 para la compensación
realFT1=calc_realFT(w1);
imagFT1=(Kp*Kd*w1-Kp*Ki/w1)*realFT1+Kp*imagFT1;

if(tipoNL==9)
  imagFT1=imagFT1+pi*D/(4*U);
w2=inicio*1000;

imagFT2=calc_imagFT(w2);

//Obtención de imagFT2 para la compensación
realFT2=calc_realFT(w2);
imagFT2=(Kp*Kd*w2-Kp*Ki/w2)*realFT2+Kp*imagFT2;

if(tipoNL==9)
  imagFT2=imagFT2+pi*D/(4*U);

if(ImagFT2*imagFT1<0)
  {
    do
      {
       w3=(w1+w2)/2;
       imagFT3=calc_imagFT(w3);

       //Obtención de imagFT3 para la compensación
       realFT3=calc_realFT(w3);
       imagFT3=(Kp*Kd*w3-Kp*Ki/w3)*realFT3+Kp*imagFT3;

       if(tipoNL==9)
         imagFT3=imagFT3+pi*D/(4*U);

       if(imagFT2*imagFT3<=0)
         {
           w1=w3;
           imagFT1=imagFT3;
         }
       else
         {
           if(imagFT1*imagFT3<=0)
             {
              w2=w3;
              imagFT2=imagFT3;
             }
         }
      }while(fabs(w1-w2)>0.0001);
    ptocFT=1;
    Wc=w1;
  }

if(ptocFT==1)
  {
    // Cálculo del punto mínimo o máximo de la función descriptiva
    minFD=realFD[0];
    for(i=1; i<60; i++)
      {
       if(minFD<realFD[i])
         minFD=realFD[i];
      }

    if(minFD==realFD[0])       //se buscará el máximo
      {
       for(i=1; i<60; i++)
         {
           if(minFD>realFD[i])
             minFD=realFD[i];
         }
```

```
  }

realFTc=calc_realFT(Wc);

//Obtención de realFTc para la compensación
imagFTc=calc_imagFT(Wc);
realFTc=Kp*realFTc+(Kp*Ki/Wc-Kp*Kd*Wc)*imagFTc;


//calculo del primer pto de cruce
A1=inicioAFD;
realFD1=calc_realFD(A1)-realFTc;
for(i=0; i<60; i++)
  {
   if(realFD[i]==minFD)
     {
       if(i<=20)
         A2=0.3*i+inicioAFD;
       else
         A2=2*i-34+inicioAFD;
     }
  }
realFD2=minFD-realFTc;

if(realFD2*realFD1<0)
  {
   do
     {
       A3=(A1+A2)/2;
       realFD3=calc_realFD(A3)-realFTc;
       if(realFD2*realFD3<=0)
         {
          A1=A3;
          realFD1=realFD3;
         }
       else
         {
          if(realFD1*realFD3<=0)
            {
              A2=A3;
              realFD2=realFD3;
            }
         }
     }while(fabs(A1-A2)>0.0001);
   Ptocorte=1;
   Ac1=A1;
  }


//calculo del segundo pto de cruce
for(i=0; i<60; i++)
  {
   if(realFD[i]==minFD)
     {
       if(i<=20)
         A1=0.3*i+inicioAFD;
       else
         A1=2*i-34+inicioAFD;
     }
  }
realFD1=minFD-realFTc;
A2=A1+1000;
realFD2=calc_realFD(A2)-realFTc;

if(realFD2*realFD1<0)
  {
   do
     {
       A3=(A1+A2)/2;
```

```
            realFD3=calc_realFD(A3)-realFTc;
            if(realFD2*realFD3<=0)
              {
                A1=A3;
                realFD1=realFD3;
              }
            else
              {
                if(realFD1*realFD3<=0)
                  {
                    A2=A3;
                    realFD2=realFD3;
                  }
              }
          }while(fabs(A1-A2)>0.0001);
        Ptocorte=1;
        Ac2=A1;
      }
    }
  }
else
  {
    w1=inicio+0.001;
    A1=inicioAFD+0.002;
    for(i=1;i<30;i++)
      {
        realFT1=calc_realFT(w1);

        //Obtención de realFT1 para la compensación
        imagFT1=calc_imagFT(w1);
        realFT1=Kp*realFT1+(Kp*Ki/w1-Kp*Kd*w1)*imagFTc;

        imagFT1=calc_imagFT(w1);

        //Obtención de imagFT1 para la compensación
        realFT1=calc_realFT(w1);
        imagFT1=(Kp*Kd*w1-Kp*Ki/w1)*realFT1+Kp*imagFT1;


        if(A1>inicioAFD)
          {
            realFD1=calc_realFD(A1);
            imagFD1=calc_imagFD(A1);
          }
        fWA=realFT1-realFD1;
        gWA=imagFT1-imagFD1;

        realFT2=calc_realFT(w1-0.001);

        //Obtención de realFT2 para la compensación
        imagFT2=calc_imagFT(w1-0.001);
        realFT2=Kp*realFT2+(Kp*Ki/(w1-0.001)-Kp*Kd*(w1-0.001))*imagFT2;

        realFT3=calc_realFT(w1+0.001);

        //Obtención de realFT3 para la compensación
        imagFT3=calc_imagFT(w1+0.001);
        realFT3=Kp*realFT3+(Kp*Ki/(w1+0.001)-Kp*Kd*(w1+0.001))*imagFT3;

        fW=(realFT3-realFT2)/0.002;


        imagFT2=calc_imagFT(w1-0.001);

        //Obtención de ImagFT2 para la compensación
        realFT2=calc_realFT(w1-0.001);
        imagFT2=(Kp*Kd*(w1-0.001)-Kp*Ki/(w1-0.001))*realFT2+Kp*imagFT2;
```

```
        imagFT3=calc_imagFT(w1+0.001);

        //Obtención de imagFT3 para la compensación
        realFT3=calc_realFT(w1+0.001);
        imagFT3=(Kp*Kd*(w1+0.001)-Kp*K1/(w1+0.001))*realFT3+Kp*imagFT3;

        gW=(imagFT3-imagFT2)/0.002;

        if(A1>inicioAFD)
          {
           realFD2=-calc_realFD(A1-0.001);
           realFD3=-calc_realFD(A1+0.001);
           fA=(realFD3-realFD2)/0.002;

           imagFD2=-calc_imagFD(A1-0.001);
           imagFD3=-calc_imagFD(A1+0.001);
           gA=(imagFD3-imagFD2)/0.002;
          }

        denom = gA*fW-gW*fA;
        w2 = w1-(gA*fWA-fA*gWA)/denom;
        A2 = A2-(fW*gWA-gW*fWA)/denom;


        if(fabs(w2-w1)<0.0001 && fabs(A2-A1)<0.0001)
          {
           Ptocorte=1;
           Wc=w2;
           Ac1=A2;
           i=30;
          }

        if(fabs(w2)>5000 || fabs(A2)>5000)
          i=30;

        w1=w2;
        A1=A2;
      }
    }

}
```

```
/*-------------------------------------------------------*/
/*                                                       */
```

```
/*   PROGRAMA PARA ANALIZAR ESTABILIDAD                          */
/*   UTILIZANDO LA FUNCION DESCRIPTIVA                           */
/*                                                               */
/*        GRAFICAR EN EL TIEMPO                                  */
/*                                                               */
/*   Wilmer Cárdenas Burbano 1994                                */
/*---------------------------------------------------------------*/


#define Uses_TApplication

#include <tv.h>
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#include <ctype.h>

#include "funcion.h"

extern int gradnum, gradden;
extern float numerador[];
extern float denominador[];

extern int factor_eje;

extern int tipoNL;
extern int hayFT;
extern int hayGc;
extern float Yk[];
extern float T;
extern float Kp, Kd, Ki;
extern float pto1, pto2, pto3, pto4;



void TVAesfudes::trazoejesTiempo()
{
   int i, j, k, xmax, ymax, xo, yo;
   static char *num_graf1[]={"","1","2","3","4","5","6","7","8","9","10",
                             "11","12","13","14"};
   static char *num_graf2[]={"2","4","6","8","10","12","14","16","18",
                          "20","22","24","26","28","30","32","34","36",
                          "38","40","42","44","46","48","50","52","54","56"};

   cleardevice();
   xmax = getmaxx();
   ymax = getmaxy();
   xo=xmax/10;
   yo=ymax-ymax/3;

    /*TITULO*/
    setcolor(getmaxcolor());
    settextjustify(1,1);
    outtextxy(xmax/2,5,"PROGRAMA SEMIGRAFICO PARA ANALIZAR");
    outtextxy(xmax/2,15,"ESTABILIDAD USANDO LA FUNCION DESCRIPTIVA");

    setcolor(6);
    setlinestyle(0,0,0);
    line(0,yo, xmax, yo);
    line(xo,0,xo,ymax);

    if(factor_eje>=200)
      {
       /*GRAFICO DE LOS MINOR TICKS*/
       for(i=0;i<xmax-xo;i+=factor_eje/50)       //ejeX
         line(i+xo,yo,i+xo,yo+3);
      }

   for(i=0;i<yo;i+=factor_eje/10)           //ejeY
```

```
    line(xo,-i+yo,xo+3,-i+yo);

  for(i=0;i<yo;i+=factor_eje/10)            //ejeY -
    line(xo,i+yo,xo+3,i+yo);


  /*GRAFICO DE LOS MEDIUMTICKS*/
  for(i=0;i<xmax-xo;i+=factor_eje/10)       //ejeX
    line(i+xo,yo,i+xo,yo+6);

  for(i=0;i<yo;i+=factor_eje/2)             //ejeY
    line(xo,-i+yo,xo+6,-i+yo);

  for(i=0;i<yo;i+=factor_eje/2)             //ejeY -
    line(xo,i+yo,xo+6,i+yo);


  if(factor_eje<200)
    {
     /*GRAFICO DE LOS MAYORTICKS*/
     j=0;
     k=2*factor_eje/5;
     for(i=0;i<xmax-xo;i+=factor_eje/5)       //ejeX
       {
         line(i+xo,yo,i+xo,yo+9);
         if(i--k)
           {
            outtextxy(i+xo,yo+20,num_graf2[j]);
            k+=2*factor_eje/5;
            j++;
           }
       }
    }
  else
    {
     j=0;
     for(i=0;i<xmax-xo;i+=factor_eje/5)       //ejeX
       {
         line(i+xo,yo,i+xo,yo+9);
         outtextxy(i+xo,yo+20,num_graf1[j]);
         j++;
       }
    }


  j=0;
  for(i=0;i<yo;i+=factor_eje)               //ejeY
    {
     line(xo,-i+yo,xo+9,-i+yo);
     outtextxy(xo+20,-i+yo,num_graf1[j]);
     j++;
    }

  j=0;
  for(i=0;i<yo;i+=factor_eje)               //ejeY -
    {
     line(xo,i+yo,xo+9,i+yo);
     outtextxy(xo+20,i+yo,num_graf1[j]);
     j++;
    }
}



void TVAesfudes::calculoTiempo()
{
  int i, j, k, ko, lo;
```

```
float b1=0, b2=0, b3=0;
float U, D, D1, D2, D3, D4, K, K2;
float dato;
char ch;

float uk, uk_1, uk_2, mk, mk_1,nk, Y;
int interv=25;
int totptos=20000;

static float A[11][11];
static float AD[11][11];
static float B[11];
static float BD[11];
static float C[11];
static float Xk[11];
static float X[11];

//INICIALIZACION DE MATRICES
for(i=0;i<gradden;i++)
  for(j=0;j<gradden;j++)
    {
     A[i][j]=0;
     AD[i][j]=0;
    }

for(i=0;i<gradden;i++)
  {
    B[i]=0;
    BD[i]=0;
    C[i]=0;
    Xk[i]=0;
    X[i]=0;
  }


for(i=0;i<400;i++)
  Yk[i]=0;


ko=0;
lo=interv;

uk=0;
uk_1=0;
uk_2=0;
mk=0;
mk_1=0;
nk=0;


setcolor(YELLOW);
gotoxy(30,25);
puts("El tiempo máximo a graficar es 20000*T segundos");
gotoxy(5,8);
puts("Ingrese el periodo de Muestreo T");
do
  {
    gotoxy(1,10);
    puts("                                          ");
    gotoxy(45,8);
    puts("                  ");
    puts("                                                \n");
    puts("                                               ");
    gotoxy(45,8);
    while(scanf("%f",&T)==0)
     {
       fflush(stdin);
       gotoxy(45,8);
       printf("\a       ");
```

```
      gotoxy(1,9);
      puts("                                    ");
      gotoxy(45,8);
    }
  gotoxy(45,8);
  printf("%f",T);
  gotoxy(1,9);
  puts("                                    ");

  setcolor(RED);
  printf("\nLOS DATOS SON CORRECTOS s/n\t");
  setcolor(YELLOW);
  fflush(stdin);
  ch=getch();
  if(isupper(ch))
    ch=tolower(ch);
  if(ch!='s')
    puts("\a");
  if(T>0.1 || T<0.00001)
    {
      gotoxy(1,12);
      setcolor(LIGHTRED);
      puts("\aT debe estar entre 0.00001<T<0.1");
      setcolor(YELLOW);
      ch='n';
    }
 }while(ch!='s');

b1 = Kp + Ki*Kp*T/2 + Kd*Kp/T;
b2 = -Kp + Ki*Kp*T/2 - 2*Kd*Kp/T;
b3 = Kd*Kp/T;


//INGRESO DE LAS CONDICIONES INICIALES
gotoxy(5,10);
puts("Ingrese las condiciones inciales (Vector X)");
do
 {
   gotoxy(1,11);
   puts("                                          ");
   gotoxy(1,12);
   puts("                                          ");
   gotoxy(1,13);
   puts("                                          ");
   gotoxy(1,14);
   puts("                                          ");
   gotoxy(1,15);
   puts("                                          ");
   gotoxy(1,16);
   puts("                                          ");
   gotoxy(1,17);
   puts("                                          ");
   gotoxy(1,18);
   puts("                                          ");
   gotoxy(1,19);
   puts("                                          ");
   gotoxy(1,20);
   puts("                                          ");
   gotoxy(1,21);
   puts("                                          ");


   for(i=0;i<gradden;i++)
    {
      gotoxy(5,12+i);
      printf("X[%d]= ",i);
      fflush(stdin);
      gotoxy(15,12+i);
```

```
        while(scanf("%f",&dato)==0)
          {
            fflush(stdin);
            gotoxy(15,12+i);
            printf("\a       ");
            gotoxy(1,12+i+1);
            puts("                                  ");
            gotoxy(1,12+i+2);
            puts("                                  ");
            gotoxy(15,12+i);
          }
        gotoxy(15,12+i);
        printf("%f",dato);
        X[i]=dato;
        puts("                                  ");
        puts("                                  ");
        puts("                                  ");
        puts("                                  ");
      }
    gotoxy(5,12+gradden+1);
    setcolor(RED);
    puts("LOS DATOS SON CORRECTOS s/n\t");
    setcolor(YELLOW);
    fflush(stdin);
    ch=getch();
    if(isupper(ch))
    ch=tolower(ch);

    if(ch!='s')
     puts("\a");

  }while(ch!='s');



//DETERMINACION DE LAS MATRICES A, B y C DE LA PLANTA

for(i=0; i<gradden-1; i++)
  A[i][i+1] = 1;
for(i=0; i<gradden; i++)
  A[gradden-1][i] = -denominador[i]/denominador[gradden];

B[gradden-1]=1;

for(i=0; i<gradden; i++)
  C[i] = numerador[i]/denominador[gradden];



//DETERMINACION DE LAS MATRICES AD y BD  DE LA PLANTA

for (i=0; i<gradden; i++)
  for(j=0; j<gradden; j++)
    AD[i][j]=A[i][j]*T;

for(i=0; i<gradden; i++)
  for(j=0; j<gradden; j++)
    if(i==j)
      AD[i][j]=AD[i][j]+1;

for(i=0; i<gradden; i++)
  BD[i]=B[i]*T;


gotoxy(40,25);
setcolor(MAGENTA);
cleardevice();
puts("ESPERE UN MOMENTO ......");
```

```
Y=0;
//LAZO DE CALCULO DE RESPUESTA EN EL TIEMPO
for(k=1;k<totptos;k++)
  {
    uk = 1-Y;
    if(hayGc==1)
      mk = mk_1 + b1*uk + b2*uk_1 + b3*uk_2;
    else
      mk=uk;

    if(tipoNL!=0)
      {
        if(tipoNL==1)
          {
            U = pto1;
            if(mk>=0)
              nk=U;
            else
              nk=-U;
          }

        if(tipoNL==2)
          {
            U=pto1;
            D=pto2;
            if(mk>=D)
              nk=U;
            if(mk>=-D && mk<D)
              nk=0;
            if(mk<-D)
              nk=-U;
          }

        if(tipoNL==3)
          {
            U=pto1;
            K=pto2;
            if(mk>=0)
              nk=K*mk+U;
            else
              nk=K*mk-U;
          }

        if(tipoNL==4)
          {
            U=pto1;
            D=pto2;
            K=pto3;
            if(mk>=D)
              nk=K*(mk-D)+U;
            if(mk>=-D && mk<D)
              nk=0;
            if(mk<-D)
              nk=K*(mk+D)-U;
          }

        if(tipoNL==5)
          {
            U=pto1;
            D=pto2;
            K=U/D;
            if(mk>=D)
              nk=U;
            if(mk>=-D && mk<D)
              nk=K*mk;
            if(mk<-D)
              nk=-U;
```

```
  }

if(tipoNL==6)
  {
    U=pto1;
    D1=pto2;
    D2=pto3;
    K=U/(D2-D1);
    if(mk>=D2)
     nk=U;
    if(mk>=D1 && mk<D2)
     nk=K*(mk-D1);
    if(mk>=-D1 && mk<D1)
     nk=0;
    if(mk>=-D2 && mk<-D1)
     nk=K*(mk+D1);
    if(mk<-D2)
     nk=-U;
  }

if(tipoNL==7)
  {
    U=pto1;
    D=pto2;
    K2=pto3;
    K=U/D;
    if(mk>=D)
     nk=K2*(mk-D)+U;
    if(mk>=-D && mk<D)
     nk=K*mk;
    if(mk<-D)
     nk=K2*(mk+D)-U;
  }

if(tipoNL==8)
  {
    U=pto1;
    D1=pto2;
    D2=pto3;
    K2=pto4;
    K=U/(D2-D1);
    if(mk>=D2)
     nk=K2*(mk-D2)+U;
    if(mk>=D1 && mk<D2)
     nk=K*(mk-D1);
    if(mk>=-D1 && mk<D1)
     nk=0;
    if(mk>=-D2 && mk<-D1)
     nk=K*(mk+D1);
    if(mk<-D2)
     nk=K2*(mk+D2)-U;
  }

if(tipoNL==9)
  {
    U=pto1;
    D=pto2;
    if(mk>=D)
     nk=U;
    if(mk>=-D && mk<D && mk<=mk_1)
     nk=U;
    if(mk>=-D && mk<D && mk>mk_1)
     nk=-U;
    if(mk<-D)
     nk=-U;
  }

if(tipoNL==10)
  {
```

```
    U=pto1;
    D1=pto2;
    D2=pto3;
    if(mk>=D2)
      nk=U;
    if(mk>=D1 && mk<D2 && mk<=mk_1)
      nk=U;
    if(mk>=D1 && mk<D2 && mk>mk_1)
      nk=0;
    if(mk>=-D1 && mk<D1)
      nk=0;
    if(mk>=-D2 && mk<=-D1 && mk<=mk_1)
      nk=0;
    if(mk>=-D2 && mk<=-D1 && mk>mk_1)
      nk=-U;
    if(mk<-D2)
      nk=-U;
  }

if(tipoNL==11)
  {
    U=pto1;
    D1=pto2;
    D2=pto3;
    K=2*U/(D2+D1);
    if(mk>=D2)
      nk=U;
    if(mk>=D1 && mk<D2 && mk<=mk_1)
      nk=U;
    if(mk>=D1 && mk<D2 && mk>mk_1)
      nk=K*(mk-D2)+U;
    if(mk>=-D1 && mk<D1 && mk<=mk_1)
      nk=K*(mk+D2)-U;
    if(mk>=-D1 && mk<D1 && mk>mk_1)
      nk=K*(mk-D2)+U;
    if(mk>=-D2 && mk<=-D1 && mk<=mk_1)
      nk=K*(mk+D2)-U;
    if(mk>=-D2 && mk<=-D1 && mk>mk_1)
      nk=-U;
    if(mk<-D2)
      nk=-U;
  }

if(tipoNL==12)
  {
    U=pto1;
    D1=pto2;
    D2=pto3;
    D3=pto4;
    D4=D2-D1+D3;
    K=U/(D2-D1);
    if(mk>=D4)
      nk=U;
    if(mk>=D3 && mk<D4 && mk<=mk_1)
      nk=U;
    if(mk>=D3 && mk<D4 && mk>mk_1)
      nk=K*(mk-D3);
    if(mk>=D2 && mk<D3 && mk<=mk_1)
      nk=U;
    if(mk>=D2 && mk<D3 && mk>mk_1)
      nk=0;
    if(mk>=D1 && mk<D2 && mk<=mk_1)
      nk=K*(mk-D1);
    if(mk>=D1 && mk<D2 && mk>mk_1)
      nk=0;
    if(mk>=-D1 && mk<D1)
      nk=0;
    if(mk>=-D2 && mk<=-D1 && mk<=mk_1)
      nk=0;
```

```
            if(mk>=-D2 && mk<-D1 && mk>mk_1)
              nk=K*(mk+D1);
            if(mk>=-D3 && mk<-D2 && mk<=mk_1)
              nk=0;
            if(mk>=-D3 && mk<-D2 && mk>mk_1)
              nk=-U;
            if(mk>=-D4 && mk<-D3 && mk<=mk_1)
              nk=K*(mk+D3);
            if(mk>=-D4 && mk<-D3 && mk>mk_1)
              nk=-U;
            if(mk<-D4)
              nk=-U;
          }
        }
      else
        nk=mk;

      Y=0;
      for(i=0;i<gradden;i++)
        Y=Y+C[i]*X[i];

      for(i=0; i<gradden; i++)
        {
          for(j=0; j<gradden; j++)
            Xk[i]=Xk[i]+AD[i][j]*X[j];

          Xk[i]=Xk[i]+BD[i]*nk;
        }

      for(i=0;i<gradden;i++)
        X[i]=Xk[i];

      for(i=0;i<gradden;i++)
        Xk[i]=0;

      if(k==lo)
        {
          ko++;
          Yk[ko]=Y;
          lo=lo+interv;
        }

      mk_1 = mk;
      uk_2 = uk_1;
      uk_1 = uk;

    }
}



int TVAesfudes::graficTiempo()
{
  char ch;
  int i, j, xmax, ymax, xo, yo;
  float x2, y2;

  xmax = getmaxx();
  ymax = getmaxy();
  xo=xmax/10;
  yo=ymax-ymax/3;

  setcolor(getmaxcolor());
  gotoxy(15,4);
  puts("      t[s]      Y[t]");

  for(i=0;i<800-1;i++)
```

```
    {
      x2=T*25*(i+1)*factor_eje/5;
      y2=Yk[i+1]*factor_eje;
      gotoxy(15,5);
      if(i<-2)
        {
          printf("FT  %9.3e  %9.3e\r", T*25*i, Yk[i]);
          line(xo+T*25*i*factor_eje/5, yo-Yk[i]*factor_eje, xo+x2, yo-y2);
        }
      else
        {
          if(xo+x2>-0 && xo+x2<-xmax && yo-y2>-0 && yo-y2<ymax)
            {
              printf("FT  %9.3e  %9.3e\r", T*25*i, Yk[i]);
              line(xo+T*25*i*factor_eje/5, yo-Yk[i]*factor_eje, xo+x2, yo-y2);
            }
        }
    }
  gotoxy(15,6);
  printf("T= %f",T);


  /*MENSAJE CAMBIAR DE ESCALA OPCIONAL*/
  setcolor(6);
  outtextxy(450,50,"CAMBIAR DE ESCALA");
  outtextxy(450,70,"+ Aumentar escala");
  outtextxy(450,90,"- Disminuir escala");

  /*CAMBIAR DE ESCALA*/
  do
    {
      j=1;
      i=0;

      setcolor(15);
      fflush(stdin);
      ch=getch();
      if(ch=='+' || ch=='-')
        {
          i=1;
          if(ch=='+')
            {
              factor_eje*=2;
              if(factor_eje>1600)
                {
                  gotoxy(1,1);
                  puts("\a");
                  factor_eje=1600;
                  j=0;
                }
            }
          else
            {
              factor_eje/=2;
              if (factor_eje<50)
                {
                  gotoxy(1,1);
                  puts("\a");
                  factor_eje=50;
                  j=0;
                }
            }
        }
    }while(j--0);
  return i;
}
```